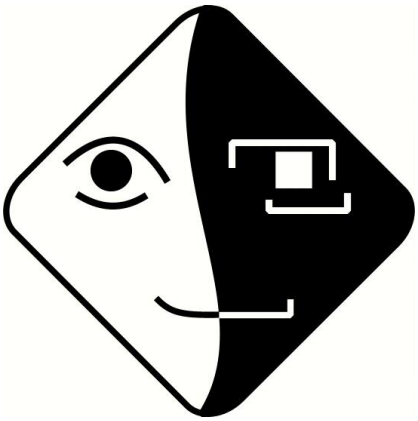
IX. Magyar Számítógépes Nyelvészeti Konferencia



MSZNY 2013

Szerkesztette:

Tanács Attila
Vincze Veronika

Szeged, 2013. január 7-8.
http://www.inf.u-szeged.hu/mszny2013

ISBN 978-963-306-189-3

Szerkesztette: Tanács Attila és Vincze Veronika
{tanacs, vinczev}@inf.u-szeged.hu

Felelős kiadó: Szegedi Tudományegyetem, Informatikai Tanszékcsoport
6720 Szeged, Árpád tér 2.

Nyomtatta: JATEPress
6722 Szeged, Petőfi Sándor sugárút 30–34.

Szeged, 2012. december

Előszó

2013. január 7-8-án kilencedik alkalommal rendezzük meg Szegeden a Magyar Számítógépes Nyelvészeti Konferenciát. A konferencia fő célja – a hagyományokhoz hűen – a nyelv- és beszédtechnológia területén végzett legújabb, illetve folyamatban levő kutatások eredményeinek ismertetése és megvitatása, mindemellett lehetőség nyílik különféle hallgatói projektek, illetve ipari alkalmazások bemutatására is. A korábbi évekhez hasonlóan, a rendezvény fokozott érdeklődést váltott ki az ország nyelv- és beszédtechnológiai szakembereinek körében.

A konferenciafelhívásra szép számban beérkezett tudományos előadások közül a programbizottság 42-t fogadott el az idei évben, így 26 előadás és 16 poszter-, illetve laptopos bemutató gazdagítja a konferencia programját. A programban a magyar számítógépes nyelvészet rendkívül széles skálájáról találhatunk előadásokat a beszédtechnológiától kezdve a számítógépes morfológia és szintaxis területén át az információkinyerésig és gépi fordításig.

Nagy örömet jelent számomra az is, hogy Gósy Mária, a Nyelvtudományi Intézet Fonetikai Osztályának tudományos osztályvezetője, az ELTE BTK Fonetika Tanszékének tanszékvezető egyetemi tanára elfogadta meghívásunkat, és *Spontán beszéd: szabályok és szabálytalanságok* című plenáris előadása is a konferenciaprogram részét képezi.

Ahogy az már hagyománnyá vált, idén is tervezzük a „Legjobb Ifjú Kutatói Díj” odaítélését, mellyel a fiatal korosztály tagjait kívánjuk ösztönözni arra, hogy kiemelkedő eredményekkel járuljanak hozzá a magyarországi nyelv- és beszédtechnológiai kutatásokhoz. A díj felajánlásáért az MTA Számítástechnikai és Automatizálási Kutatóintézetének tartozunk köszönettel.

Szeretnék köszönetet mondani a programbizottságnak: Vámos Tibor programbizottsági elnöknek, valamint Alberti Gábor, Gordos Géza, Kornai András, László János, Prószéky Gábor és Váradi Tamás programbizottsági tagoknak. Szeretném továbbá megköszönni a rendezőbizottság és a kötetszerkesztők munkáját is.

Csirik János, a rendezőbizottság elnöke

Szeged, 2012. december

Tartalomjegyzék

I. Beszédtechnológia, fonológia

II. Lexikológia, fordítás

III. Korpusznyelvészet

IV. Pszichológia

V. Morfológia, szintaxis

VI. Szemantika

VII. Információkinyerés és -visszakeresés

VIII. Poszterek és laptopos bemutatók

I. Beszédtechnológia, fonológia

Mély neuronhálók az akusztikus modellezésben

Grósz Tamás, Tóth László*

MTA-SZTE Mesterséges Intelligencia Kutatócsoport,
e-mail: groszt@sol.cc.u-szeged.hu,tothl@inf.u-szeged.hu

Kivonat A beszédfelismerők akusztikus modelljeként az utóbbi években jelentek meg, és egyre nagyobb népszerűségnek örvendenek az ún. mély neuronhálók. Nevüket onnan kapták, hogy a korábban szokványos egyetlen rejtett réteg helyett jóval többet, 3-9 réteget használnak. Emiatt – bár a hagyományos módszerekkel is taníthatók – az igazán jó eredmények eléréséhez egy új tanítóalgoritmust is ki kellett hozzájuk találni. Cikkünkben röviden bemutatjuk a mély neuronhálók matematikai hátterét, majd a mély neuronhálókra épülő akusztikus modelleket beszédhang-felismerési teszteken értékeljük ki. Az eredményeket összevetjük a korábban publikált, hagyományos neuronhálót használó eredményeinkkel.

Kulcsszavak: mély neuronháló, akusztikus modellezés, beszédfelismerés

1. Bevezetés

Az elmúlt néhány évtizedben a mesterséges neuronhálók számos változatát kipróbálták a beszédfelismerésben - annak függvényében, hogy éppen mi volt az aktuálisan felkapott technológia. Általános elismertséget azonban csak a többrétegű perceptron-hálózatokra (MLP) épülő ún. hibrid HMM/ANN modellnek sikerült elérnie, főleg a Bourlard-Morgan páros munkásságának köszönhetően [1]. Bár kisebb felismerési feladatokon a neuronhálós modellek jobb eredményt adnak, mint a sztenderd rejtett Markov-modell (HMM), alkalmazásuk mégsem terjedt el általánosan, részben mivel technikailag nehézkesebb a használatuk, másrészt mivel nagyobb adatbázisokon az előnyük elvész, köszönhetően a HMM-ekhez kifejlesztett trifón modellezési és diszkriminatív tanítási technikáknak. Így a hibrid modell az elmúlt húsz évben megmaradt a versenyképes, de igazi áttörést nem hozó alternatíva státuszában.

Mindez megváltozni látszik azonban az ún. mély neuronhálók (deep neural nets) megjelenésével. A mély neuronhálót (pontosabban tanítási algoritmusát) 2006-ban publikálták először [2], és a kezdeti cikkek képi alakfelismerési teszteket használtak demonstrációként. Legjobb tudomásunk szerint a mély hálók első beszédfelismerési alkalmazása Mohamed 2009-es konferenciaanyaga volt (ennek

* Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

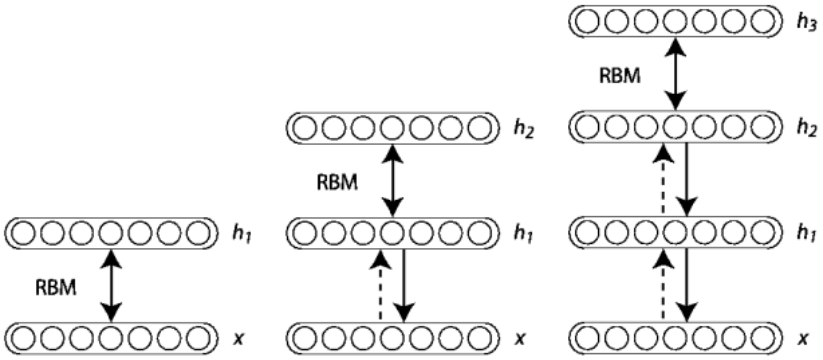
[3] az újságcikké kibővített változata) – mely cikkben rögtön sikerült megdönteni a népszerű TIMIT benchmark-adatbázison elért összes korábbi felismerési pontosságot. A modellt ráadásul hamarosan tovább javították [4]-ben. Ezek az eredmények annyira meggyőzőek voltak, hogy azóta exponenciálisan nő a témával foglalkozó cikkek száma - a legutóbbi, 2012. szeptemberi Interspeech konferencián már két szekció volt speciálisan csak a mély neuronhálóknak szentelve.

Cikkünkben először bemutatjuk a mély neuronhálók matematikai hátterét. Kitérünk a betanításuk során használt korlátos Boltzmann-gépekre, illetve a ,,kontrasztív divergencia" elnevezésű tanító algoritmusukra. A kísérleti alátámasztásra beszédhang-felismerési teszteket végzünk három adatbázison. Az angol nyelvű TIMIT-en megkíséreljük reprodukálni a [3]-ben közölt eredményeket, majd pedig két magyar nyelvű korpuszra – egy híradós adatbázis és egy hangoskönyv – terjesztjük ki a vizsgálatokat. Mindkét adatbázison közöltünk már eredményeket korábban, ezek fogják képezni a kiértékelés viszonyítási pontját.

2. Mély neuronhálók

Miben is különbözik ez az új neuronhálós technológia a megszokott többrétegű perceptronoktól? Egyrészt a hálózat struktúrájában, másrészt a tanító algoritmusban. A hagyományos hálózatok esetében egy vagy maximum két rejtett réteget szoktunk csak használni, és a neuronok számának növelésével próbáljuk a hálózat osztályozási pontosságát növelni. Emellett az az elméleti eredmény szól, miszerint egy kétrétegű hálózat már univerzális approximátor, azaz egy elég általános függvényosztályon tetszőleges pontosságú közelítésre képes [5]. Ehhez azonban a neuronok számát tetszőleges mértékben kell tudni növelni. Ehhez képest az újabb matematikai érvek és az empirikus kísérletek is amellett szólnak, hogy - *adott neuronszám mellett* - a több réteg hatékonyabb reprezentációt tesz lehetővé [6]. Ez indokolja tehát a sok, relatíve kisebb rejtett réteg alkalmazását egyetlen, rengeteg neuront tartalmazó réteg helyett.

Az ilyen sok rejtett réteges, „mély" architektúrának azonban nem triviális a betanítása. A hagyományos neuronhálók tanítására általában az ún. backpropagation algoritmust szokás használni, ami tulajdonképpen a legegyszerűbb, gradiensalapú optimalizálási algoritmus neuronhálókhoz igazított változata. Ez egy-két rejtett réteg esetén még jól működik, ennél nagyobb rétegszám mellett azonban egyre kevésbé hatékony. Ennek egyik oka, hogy egyre mélyebbre hatolva a gradiensek egyre kisebbek, egyre inkább "eltűnnek" (ún. „vanishing gradient" effektus), ezért az alsóbb rétegek nem fognak kellőképp tanulni [6]. Egy másik ok az ún. „explaining away" hatás, amely megnehezíti annak megtanulását, hogy melyik rejtett neuronnak mely jelenségekre kellene regálnia [2]. Ezen problémák kiküszöbölésére találták ki a korlátos Boltzmann-gépet (Restricted Boltzmann Machine, RBM), illetve annak tanító algoritmusát, a CD-algoritmust (kontrasztív divergencia) [2]. A korlátos Boltzmann-gép lényegében a neuronháló egy rétegpárjának felel meg, így a betanítás rétegenként haladva történik. A tanítás végén a rétegpárok egymásra helyezésével előálló többrétegű hálót "Deep Belief Network"-nek hívják az irodalomban [3]. Az elmondottakat szemlélteti a 1. ábra.



1. ábra. Korlátos Boltzmann-gép, illetve a belőle felépített DBN.

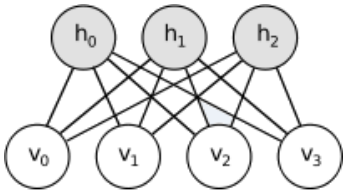
Fontos még tudni, hogy a CD-algoritmus felügyelet nélküli tanítást végez, és tulajdonképpen a „maximum likelihood" tanítás egy hatékony közelítését adja. Ezért a CD-algoritmus szerint tanítást tulajdonképpen előtanításnak tekintjük, mivel ezután következik még a címkézett tanítópéldákhoz való hozzáigazítás. E célból a hálózatot átalakítjuk korlátos Boltzmann-gépek helyett hagyományos neuronokat használó hálózattá, ráteszünk egy softmax-réteget, és ezután a megszokott backpropagation-algoritmussal végezzük a címkéken való felügyelt tanítást. A tanítás tehát két szakaszra oszlik: egyik az előtanítás, a másik pedig a hagyományos hálózatként való finomhangolás. Ha az előtanítást elhagyjuk, akkor egy teljesen hagyományos neuronhálót kapunk, így az előtanítási módszer hatékonyságának mérésére az a legjobb módszer, ha megnézzük, hogy mennyit javulnak a felismerési eredmények a használatával az előtanulást nem alkalmazó hálóhoz képest.

Az alábbi két fejezetben bemutatjuk a korlátos Boltzmann-gépeket, illetve a tanításukra szolgáló CD-algoritmust.

2.1. RBM

A korlátos Boltzmann-gép lényegében egy Markov véletlen mező (MRF), amely két rétegből áll. A korlátos jelző onnét származik, hogy két neuron csak akkor van összekapcsolva, ha az egyik a látható, a másik pedig a rejtett réteghez tartozik. Tehát a régeken belül a neuronok nem állnak kapcsolatban, ezért tekinthetünk az RBM-re úgy is, mint egy teljes páros gráf, ezt szemlélteti a 2. ábra. Az egyes kapcsolatokhoz tartozó súlyok és a neuronokhoz tartozó bias-ok egy véletlen eloszlást definiálnak a látható réteg neuronjainak állapotait tartalmazó v vektorok felett, egy energiafüggvény segítségével. Az energiafüggvény (v, h) együttes előfordulására:

$$E(v, h, \Theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j, \tag{1}$$



2. ábra. Egy RBM 4 látható és 3 rejtett neuronnal.

ahol $\Theta = (w, b, a)$, és w_{ij} reprezentálja az i. látható neuron és j. rejtett neuron szimmetrikus kapcsolatának súlyát, b_i a látható, illetve a_j pedig a rejtett neuronokhoz tartozó bias-okat. V és H a látható és rejtett egységek/neuronok száma. A modell által a v látható vektorhoz rendelt valószínűség:

$$p(v, \Theta) = \frac{\sum_h e^{-E(v,h)}}{\sum_u \sum_h e^{-E(u,h)}}, \tag{2}$$

ahol u eleme az input vektoroknak, h pedig a rejtett réteg állapotvektorainak. Mivel a korlátos Boltzmann gépben nem engedélyezett rejtett-rejtett és látható-látható kapcsolat, ezért $p(v|h)$-t és $p(h|v)$-t a következő módon definiálhatjuk:

$$\begin{aligned} p(h_j = 1|v, \Theta) &= \sigma(\sum_{i=1}^{V} w_{ij} v_i + a_j) \\ p(v_i = 1|h, \Theta) &= \sigma(\sum_{j=1}^{V} w_{ij} h_j + b_i), \end{aligned} \tag{3}$$

ahol $\sigma(x) = 1/(1 + \exp(-x))$ a szigmoid függvény.
Speciális változata az RBM-eknek az ún. Gauss-Bernoulli RBM, amely esetén a látható réteg neuronjai nem binárisak, hanem valós értékűek. Ezt valós input esetén szokás használni, és az energiafüggvény ekkor a következőképpen módosul:

$$E(v, h|\Theta) = \sum_{i=1}^{V} \frac{(v_i - b_i)^2}{2} - \sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{j=1}^{H} a_j h_j \tag{4}$$

A v látható vektorhoz rendelt valószínűség pedig:

$$p(v_i = 1|h, \Theta) = \mathcal{N}(b_i + \sum_{j=1}^{H} w_{ij} h_j, 1), \tag{5}$$

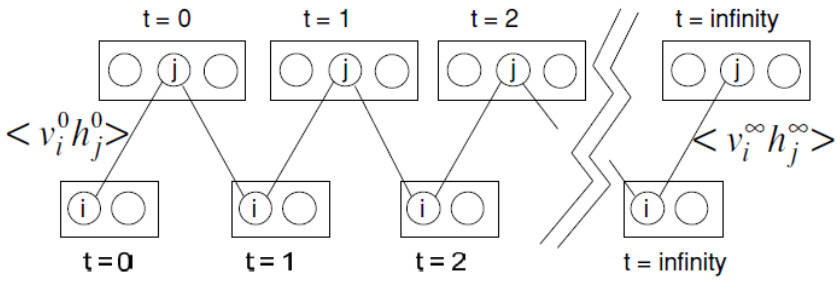
ahol $\mathcal{N}(\mu, \sigma)$ a μ várható értékű és σ varianciájú Gauss-eloszlás.
A pontos maximum likelihood tanulás alkalmatlan nagy méretű RBM esetén, ugyanis a derivált számításának időigénye exponenciálisan nő a hálózat méretével. A hatékony megoldást egy közelítő tanító algoritmus, az ún. kontrasztív divergencia (Contrastive Divergence, CD) biztosítja. Ennek a hatékony tanító algoritmusnak köszönhetően az RBM tökéletesen alkalmas arra, hogy a mély neuronhálók építőeleme legyen.

2.2. A CD-algoritmus

Hinton 2006-os cikkében javasolt egy tanító algoritmust a korlátos Boltzmann-gépekhez, amelyet kontrasztív divergenciának (Contrastive Divergence) nevezett el [2]. A javasolt módszer során a súlyok frissítési szabálya:

$$\Delta w_{ij} \propto \langle v_i h_j \rangle_{input} - \langle v_i h_j \rangle_{rekonstrukcio}. \tag{6}$$

A (6) jobb oldalán található első tag az i. látható és j. rejtett egység korrelációja, bináris esetben annak gyakorisága, hogy mindkét neuron egyszerre aktív. A rejtett réteg állapotát adott inputvektorhoz (3) alapján számítjuk. A második tag jelentése hasonló, csak ekkor rekonstrukciós állapotokat használunk. Rekonstrukció alatt a következőt kell érteni: miután az input alapján meghatároztuk a rejtett réteg állapotait, (3) felhasználásával tudjuk (a rejtett réteg alapján) a látható réteg állapotait kiszámolni, ezután az így kapott látható réteghez generáljuk a rejtett réteget. A rekonstrukciót tetszőleges alkalommal megismételhetjük a 3. ábrán látható módon.



3. ábra. Rekonstrukciós lánc.

Mivel a rekonstrukciós lépések rendkívül időigényesek, ezért általában csak k db rekonstrukciót végzünk. A CD mohó algoritmusa $k = 1$ rekonstrukciót végez, és az alapján tanulja a súlyokat, általánosan ez a módszer terjedt el viszonylag kis időigénye és jó teljesítménye miatt. A mohó előtanítás során a súlyok frissítését a következő módon végezzük:

$$\Delta w_{ij} \propto \langle v_i h_j \rangle_{input} - \langle v_i h_j \rangle_{t=1}. \tag{7}$$

Mint már korábban említettük, az előtanítás után a hálózatot átalakítjuk hagyományos neuronhálóvá, ami egyszerűen csak a súlyok átvitelével, illetve egy softmax-réteg felhelyezésével történik. Innentől a háló teljesen szokványosan tanítható felügyelt módon a backpropagation algoritmus segítségével. Mivel a tanításnak ez a része közismertnek tekinthető, ezért ennek az ismertetésétől eltekintünk.

3. Kísérleti eredmények

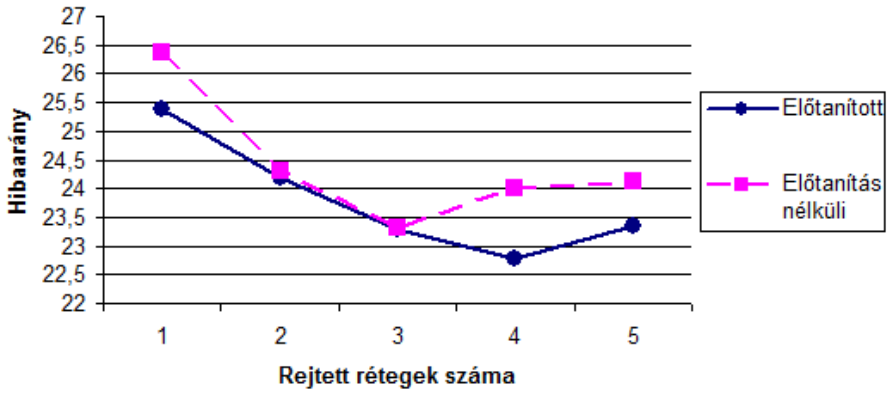
A továbbiakban kísérleti úton vizsgáljuk meg, hogy a mély neuronhálók milyen pontosságú beszédfelismerést tesznek lehetővé. Az akusztikus modellek készítése az ún. hibrid HMM/ANN sémát követi [1], azaz a neuronhálók feladata az akusztikus vektorok alapján megbecsülni a rejtett Markov-modell állapotainak valószínűségét, majd ezek alapján a teljes megfigyeléssorozathoz a rejtett Markov-modell a megszokott módon rendel valószínűségeket. Mivel a neuronhálóknak állapot-valószínűségeket kell visszaadniuk, ezért minden esetben első lépésben egy rejtett Markov-modellt tanítottunk be a HTK programcsomag használatával [7], majd ezt kényszerített illesztés üzemmódban futtatva kaptunk állapotcímkéket minden egyes spektrális vektorhoz. Ezeket a címkéket kellett a neuronhálónak megtanulnia, amihez inputként az aktuális akusztikus megfigyelést, plusz annak 7-7 szomszédját kapta meg. Az előtanítás a következő paraméterekkel történt: a tanulási ráta 0.002 volt a legalsó (Gauss-Bernoulli) rétegre, a magasabb (bináris) rétegekre 0.02. A A tanítás ún. kötegelt módon történt, ehhez a batch méretét 128-ra állítottuk, és 50 iterációt futtattunk az alsó, 20-at a többi rétegen. A backpropagation tanítás paraméterei az alábbiak voltak: a tanulási ráta 0.02-ről indult, a batch mérete ismét 128 volt. Mindegyik esetben alkalmaztuk az ún. momentum módszert, ennek paraméterét 0.9-re állítottuk.

A modellek kiértékelését háromféle adatbázison végeztük el. Mindhárom esetben azonos volt az előfeldolgozás: e célra a jól bevált mel-kepsztrális együtthatókat (MFCC) használtuk, egész pontosan 13 együtthatót (a nulladikat is beleértve) és az első-második deriváltjaikat. Közös volt még továbbá, hogy egyik esetben sem használtunk szószintű nyelvi modellt, pusztán egy beszédhangbigram támogatta a felismerést. Ennek megfelelően a felismerő kimenete is beszédhang szintű volt, ennek a hibáját (*1-accuracy*) fogjuk mérni a továbbiakban.

3.1. TIMIT

A TIMIT adatbázis a legismertebb angol nyelvű beszédadatbázis [8]. Habár mai szemmel nézve már egyértelműen kicsinek számít, a nagy előnye, hogy rengeteg eredményt közöltek rajta, továbbá a mérete miatt viszonylag gyorsan lehet kísérletezni vele, ezért továbbra is népszerű, főleg ha újszerű modellek első kiértékeléséről van szó. Esetünkben azért esett rá a választás, mert a mély neuronhálók első eredményeit is a TIMIT-en közölték [3], így kézenfekvőnek tűnt a használata az implementációnk helyességének igazolására.

A tanításhoz a szokványos tanító-tesztelő felosztást alkalmaztuk, azaz 3696 mondat szolgált tanításra és 192 tesztelésre (ez a kisebbik, ún. 'core' teszthalmaz). Az adatbázis 61 beszédhangcímkét használ, viszont sztenderdnek számít ezeket 39 címkére összevonni. Mi ezt az összevonást csupán a kiértékelés során tettük meg. Ez azt jelenti, hogy a monofón modellek tanítása során $61 \cdot 3 = 183$ címkével dolgoztunk (hangonként 3 állapot), azaz ennyi volt a neuronháló által megkülönböztetendő osztályok száma. Egy további kísérletben környezetfüggő (trifón) modelleket is készítettünk, ismét csak a HTK megfelelő eszközeit alkalmazva. Ennek eredményeként 858 állapot adódott, azaz ennyi osztályon tanítot-



4. ábra. Az előtanítás hatása a TIMIT core teszt halmazon a rejtett rétegek számának függvényében.

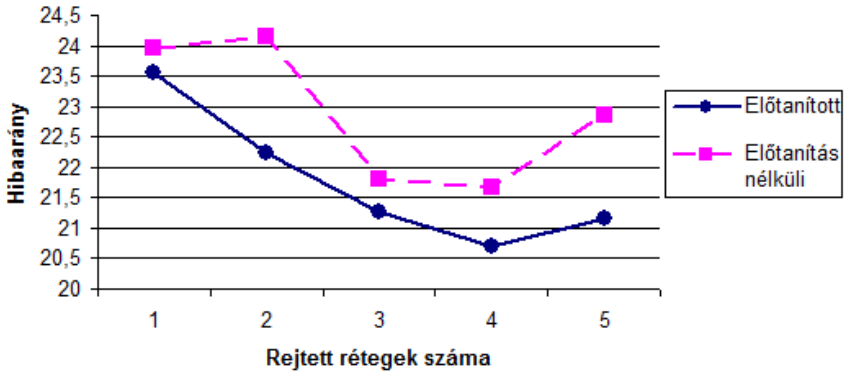
tuk a neuronhálót. A 4. ábra mutatja a monofón modellel kapott eredményeket, annak függvényében, hogy hány rejtett réteget használtunk. Az egyes rétegek neuronszáma minden esetben 1024 volt.

Az eredmények jól érzékeltetik, hogy érdemes egynél több rejtett réteget felvenni, de legfeljebb három-négyet, mert azon túl az eredmények nem javulnak számottevően (sőt, romlanak). Megfigyelhetjük továbbá, hogy az előtanítás tényleg segít, főleg mélyebb háló, azaz 4-5 réteg esetén: 4 rétegnél az eltérés az előtanítás nélküli és az előtanított háló között több mint 1% (ez kb. 5% hibacsökkenést jelent). Meg kell jegyezzük, hogy míg 4 réteg esetén az általunk kapott eredmény lényegében megegyezik az eredeti cikkben szereplővel ([3]), 5 réteg esetén nálunk már romlik az eredmény, míg ott javul. Ennek okait keressük, valószínűleg a paramétereket kell tovább hangolnunk (pl. az iterációszámot növelnünk). Azt is el kell mondanunk, hogy az itt látottaknál jobb eredményeket is el lehet érni mély neuronhálókkal (l. szintén [3]), ehhez azonban másfajta, jóval nagyobb elemszámú jellemzőkészletre van szükség. Mi most itt maradtunk az MFCC jellemzőknél, mivel ez a legáltalánosabban elfogadott jellemzőkészlet.

Rejtett rétegek száma	Hibaarány
3	22,04%
4	22,09%
5	21,91%

1. táblázat. Beszédhang-felismerési hibaarány a TIMIT adatbázison trifón címkék használata esetén.

A 1. táblázat a környezetfüggő címkékkel kapott eredményeket mutatja a TIMIT adatbázison (csak előtanításos esetre). Látható, hogy itt már öt rejtett réteg esetén kapjuk a legjobb eredményt, és az is látszik, hogy a monofón címkés eredményekhez képest kb. 1% javulás mutatkozik.



5. ábra. Az előtanítás hatása a híradós adatbázison a rejtett rétegek számának függvényében.

3.2. Híradós adatbázis

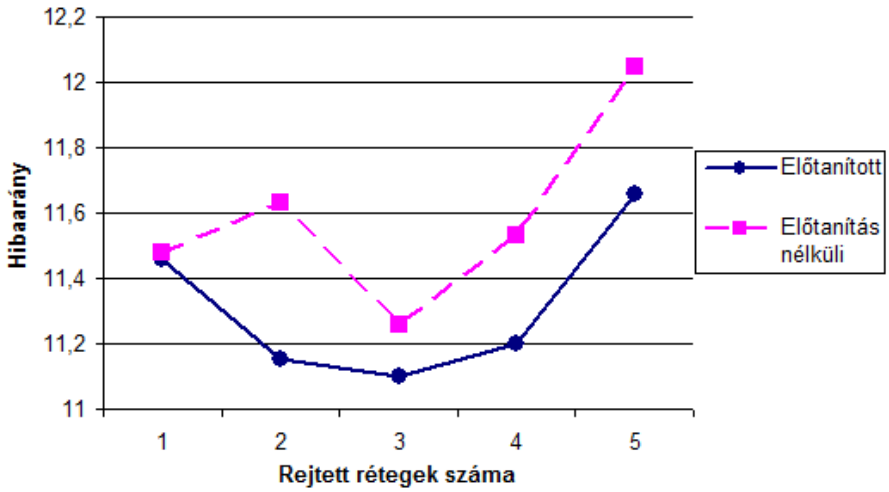
A magyar nyelvű felismerési kísérletekhez felhasznált híradós adatbázis megegyezik a [9]-ben ismertetettel. Az adatbázisnak ismét csak a a „tiszta” címkét kapott részeit használtuk fel, ami egy kb. öt és fél órás tanító és egy egyórás tesztelő részt eredményezett. Egy kétórás blokkot fenntartottunk a meta-paraméterek belövésére. Az adatbázis csak ortografikus átiratot tartalmaz, ezt egy egyszerű fonetikus átíróval alakítottuk át fonetikai címkékre, mely címkekészlet 52 elemből állt. Ebből a TIMIT adatbázisnál ismertetett módon készítettünk HMM-állapotoknak megfelelő címkézést.

A 5. ábra mutatja a monofón modellekkel elért eredményeket, különféle rétegszám mellett, ismét csak rétegenként 1024 neuronnal. Ezen az adatbázison az előtanítás kedvező hatása sokkal egyértelműbben megmutatkozik. A legjobb eredményt ismét csak négy rejtett réteggel kapjuk, a különbség az előtanítás nélküli és az előtanított rendszer között közel 1% (hibacsökkenésben kifejezve ez közel 5%). Összehasonlításképpen, korábban egy hagyományos, azaz egyetlen rejtett réteget használó hibrid modellel 23,07%-os eredményt közöltünk [9], ahhoz képest az itt szereplő 20,7% több mint 10%-os javulást jelent.

Rejtett rétegek száma	Hibaarány
3	17,94%
4	17,95%
5	18,51%

2. táblázat. Beszédhang-felismerési hibaarány a híradós adatbázison trifón címkék használata esetén

Ezen az adatbázison is megismételtük a kísérleteket környezetfüggő, azaz trifón címkékkel is (ismét csak előtanítással). Az eredmények a 2. táblázatban



6. ábra. Az előtanítás hatása a hangoskönyv-adatbázison a rejtett rétegek számának függvényében.

láthatóak. A legjobb értékeket ismét csak három és négy rejtett réteggel kaptuk, öt réteg esetén már romlás figyelhető meg. Az eredmények közel 3%-kal jobbak, mint monofón címkék esetén, ami hibacsökkenésben kifejezve 13%-os javulást jelent. Összehasonlításképp, a [9]-ben közölt legjobb trifónos korábbi eredmény 16.67% volt, tehát jobb a mostani eredménynél, de az összehasonlításhoz figyelembe kell venni, hogy ott egy ún. kétfázisú modellt alkalmaztunk, azaz két neuronháló volt egymásra tanítva, és a tanítás módja is jóval komplikáltabb volt az itt ismertetettnél. Semmi elvi akadálya nincs annak, hogy az ott közölt technológiát mély neuronhálókkal kombináljuk, ez várhatóan további javulást eredményezne.

3.3. Hangoskönyv

2009-ben beszédfelismerési kísérleteket végeztünk egy hangoskönyvvel, hogy lássuk, mit tudnak elérni a beszédfelismerők közel ideális beszédjel esetén [10]. Most ugyanazt az adatbázist vettük elő, ugyanazokkal az előkészítő lépésekkel és train-teszt felosztással. A felhasznált címkézés is ugyanaz volt.

A 6. ábra mutatja a különféle rétegszámmal elért eredményeket előtanulással és előtanulás nélkül, ismét csak rétegenként 1024 neuronnal. Érdekes módon ebben az esetben minimális volt csak az eltérés a 2-3-4 rétegszámú hálózatok eredményei között, és a legjobb eredményt három rejtett réteggel kaptuk. Az előtanulás ismét csak javított az eredményeken, de ennek hatása is kevésbé jelentős. A magyarázat valószínűleg az, hogy ez a tanulási feladat lényegesen könnyebb a másik kettőnél, és emiatt kevesebb rejtett réteg is elegendő a tanuláshoz.

Végezetül, a 3. táblázat mutatja a trifón címkézéssel kapott eredményeket. Ez esetben is a három rejtett réteges hálózat bizonyult a legjobbnak, és az

eredmények körülbelül egy százalékkal jobbak, mint a monofón címkék esetében. Ez relatív hibában kifejezve majdem tíz százalék, tehát szignifikáns javulás. Azt is elmondhatjuk továbbá, hogy az itt bemutatott eredmények lényegesen jobbak, mint a korábban tandem technológiával elért 13,16% ugyanezen adatbázison [10].

Rejtett rétegek száma	Hibaarány
3	10,24%
4	10,77%
5	11,32%

3. táblázat. Beszédhang-felismerési hibaarány a hangoskönyv-adatbázison trifón címkék használata esetén.

4. Konklúzió

Cikkünkben bemutattuk a mély neuronhálókra épülő akusztikus modelleket. A kísérleti eredmények egyértelműen igazolják, hogy a több rejtett réteg használata számottevően tud javítani az eredényeken. A „kontrasztív divergencia" előtanító algoritmus is egyértelműen hasznosnak bizonyult, bár ennek már most is sokan keresik a továbbfejlesztési lehetőségeit, főleg a nagy műveletigénye miatt. Mivel az egész témakör nagyon friss, bizonyosak lehetünk benne, hogy még számos újdonsággal fogunk találkozni e témában.

Hivatkozások

1. Bourlard, H., Morgan, N.: Connectionist Speech Recognition – A Hybrid Approach. Kluwer (1994)
2. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation, Vol. 18 (2006) 1527–1554
3. Mohamed, A., Dahl, G. E., Hinton, G.: Acoustic modeling using deep belief networks. IEEE Trans. ASLP, Vol. 20, No. 1 (2012) 14–22
4. Dahl, G. E., Ranzato, M., Mohamed, A., Hinton, G.: Phone recognition with the mean-covariance restricted boltzmann machine. In: NIPS (2010) 469–477
5. Bishop, C. M.: Pattern Recognition and Machine Learning. Springer (2006)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proc. AISTATS (2010) 249–256
7. Young, S. et al.: The HTK Book. Cambridge University Engineering Department (2005)
8. Lamel, L., Kassel, R., Seneff, S.: Speech database development: Design and analysis of the acoustic-phonetic corpus. In: Proc. DARPA Speech Recognition Workshop (1986) 121-124
9. Gosztolya G., Tóth L.: Kulcsszókeresési kísérletek hangzó híranyagokon beszédhang alapú felismerési technikákkal. In: MSZNY 2010 (2010) 224–235
10. Tóth L.: Beszédfelismerési kísérletek hangoskönyvekkel. In: MSZNY 2009 (2009) 206–216

Magyar nyelvű, kísérleti e-mail diktáló rendszer

Tarján Balázs[1], Nagy Tímea[1], Mihajlik Péter[1,2], Fegyó Tibor[1,3]

[1] Budapesti Műszaki és Gazdaságtudományi Egyetem,
Távközlési és Médiainformatikai Tanszék
{tarjanb, nagyt, mihajlik, fegyo}@tmit.bme.hu
[2] THINKTech Kutatási Központ Nonprofit Kft.
[3] AITIA International Zrt.

Kivonat: Bár a közelmúltban a szélesebb közönség számára is hozzáférhetővé váltak magyar nyelvű diktálórendszerek, használatukhoz állandó internetkapcsolat szükséges, nem teszik ki az írásjeleket és a kis-nagy kezdőbetűk használata sem követi a helyesírási szabályokat. Cikkünkben beszámolunk egy olyan diktálórendszer fejlesztéséről, mely akár a felhasználó eszközén (pl. laptop) futva, egyes írásjelek automatikus elhelyezése mellett képes számok, emotikonok, nagybetűs szavak és rövidítések felismerésére is, így drasztikus mértékben csökkentheti a bediktált szöveg utólagos gondozására fordítandó időt. Ékezetesítő eljárás használatával és a felismerő modellek személyre szabásával 26%-os szóhibaarányt értünk el nagyszótáras, e-mail diktálási feladaton. Kísérleti rendszerünkben megvizsgáltuk az egyes írásjelek automatikus elhelyezésének lehetőségeit is. Eddigi eredményeink azt mutatják, hogy csak a „vessző” kiváltására kapható megfelelően pontos előrejelzés a nyelvi modell alapján.

1 Bevezetés

Régi vágyunk, hogy magyar nyelven, viszonylag kötetlen témakörben diktálhassuk elektronikus leveleinket. Noha a közelmúltban a szélesebb közönség számára is megjelentek ilyen alkalmazások (Nuance, Google magyar nyelvű diktálórendszerek okostelefonokra), hamar szembesülniük kellett a felhasználóknak e rendszerek korlátaival. Ilyen például, hogy ezek használatához állandó internetkapcsolat szükséges, hogy a felismerési hibák kisebb-nagyobb százalékban elkerülhetetlenek, a javításuk nehézkes, továbbá nem teszik ki az írásjeleket, és a kis-nagy kezdőbetűk használata sem követi a helyesírási szabályokat. Ráadásul mindkét rendszer távoli szervereken futtatja a felismerést, mely adatvédelmi problémákat is felvethet az arra érzékeny felhasználóknál.

Cikkünkben beszámolunk egy olyan **magyar nyelvű diktálórendszer** fejlesztéséről, mely akár a felhasználó eszközén (pl. laptop) futva, egyes írásjelek automatikus elhelyezése mellett képes számok, emotikonok, nagybetűs szavak és rövidítések felismerésére is, így drasztikus mértékben csökkentheti a bediktált szöveg utólagos gondozására fordítandó időt. A pontosság növelése érdekében egy ékezetesítő eljárást is bevetettünk a tanítószöveg hibáinak javítására illetve személyre szabott felismerő modellekkel is végzünk kísérleteket. Magyar nyelvű kvázi kötetlen diktálásról igen kevés korábbi publikáció született, legjelesebb irodalomnak az [1] tekinthető. Az itt

ismertetett felismerő nagyszótáras, morfoszintaktikai szabályokkal kiegészített, morfémaalapú nyelvi modellen alapult. Fontos megjegyezni ugyanakkor, hogy e korai rendszer gyakorlati hasznát erősen korlátozta, hogy nyelvi modelljét hírlapok szövegén tanították, valamint hogy a diktálást segítő lexikai elemek sem képezték a rendszer részét. Magyar nyelvű diktálási eredményeket emellett még [2]-ben találunk, mely egy kórházi leletező rendszert mutat be. Bár folyamatos diktálásra itt is van lehetőség, azonban csupán közepes szótárméretű, szűk témájú és kis perplexitású felismerési feladaton. Ezzel szemben jelenlegi kísérleteink célja egy, a gyakorlatban is jól használható diktálóalkalmazás létrehozása volt.

2 A kezdeti nagyszótáras e-mail felismerő

Ebben a fejezetben az e-mail diktáló rendszerünk alapjául szolgáló kezdeti nagyszótáras, folyamatos beszédfelismerőt mutatjuk be. Először kitérünk a tanítóadatok begyűjtésével és feldolgozásával kapcsolatos kérdésekre, majd bemutatjuk a felismerő rendszerben használt modellek tanítási lépéseit. A fejezetet a kezdeti eredmények ismertetésével zárjuk.

2.1 Tanítóadatok gyűjtése és előfeldolgozása

Kísérleti e-mail diktáló rendszerünk tanításához olyan szöveges adatbázist kerestünk, mely elegendően nagy egy gépi beszédfelismerő nyelvi modelljének a betanításához, azonban nem tartalmaz bizalmas jellegű, személyes információkat. Ezért esett a választásunk a tanszéki laborcsoportunk belső levelezésére. További előny, hogy a betanított rendszert laborunk tagjai akár a mindennapok során is tesztelhetik, így hamarabb derülhetnek ki az esetleges hibák, és merülhetnek fel továbbfejlesztéssel kapcsolatos ötletek.

Az adatgyűjtés első lépésében a labor minden tagjától begyűjtöttük a leveleket a tanszék alapértelmezett levelezőkliensének tárolási formátumában. Ez a formátum tartalmazza a feladó, címzett, tárgy stb. mezők adatait is, melyet egyelőre a kísérleti rendszerünkben nem vettük figyelembe. A kezdeti rendszer tanításához kivettünk minden írásjelet az e-mailekből. Annak érdekében, hogy meg tudjuk jeleníteni a mondaton belüli nagybetűs szavakat, a szokásos kisbetűsítés helyett egy speciális normalizálást alkalmaztunk [3]. Minden nagybetűs szóalakot eredeti formájában hagytuk, mely alól egyedül a mondatkezdő szavak képeztek kivételt. A mondatok kezdőszavait csak akkor hagytuk meg nagybetűsnek, ha a Hunmorph [4] morfológiai elemző kizárólag ebben az alakban fogadta el őket. A [3]-ben bemutatott módszert követve a számok és a kiejtési kivételszótárban feloldott rövidítések, betűszavak felismerése is lehetővé vált. Minta a kezdeti rendszer tanítószövegének egy sorára:

```
„a Redmine-on keresetem a VOXerver dokumentációját de
végül nem találtam meg”
```

2.2 Tanítás és dekódolás

A kezdeti felismerő nyelvi modelljének tanításához egy összesen 4 millió szót tartalmazó e-mail korpuszt használtunk fel. A nyelvi modellek – mint minden további kísérleteinkben szereplő modell – módosított Kneser-Ney simítás [5] használatával készültek az SRI Language Modeling Toolkit (**SRILM**) [6] segítségével. A létrehozott 3-gram, szóalapú modellekben entrópiaalapú metszést egyetlen esetben sem alkalmaztuk.

Az e-mail diktálási feladathoz szorosan illeszkedő hanganyag előzetesen nem állt rendelkezésünkre, így egy, a feladattól független akusztikus modellt kellett használnunk a kezdeti rendszerben. A Egri Katolikus Rádió (EKR) beszélgetéseiből válogatott, összesen 43 óra hanganyagon tanított, környezetfüggő akusztikus modell a Hidden Markov Model Toolkit [7] eszközeinek segítségével készült, és összesen 6121 egyenként 13 Gauss-függvényből álló állapotot tartalmaz.

A 16 kHz-en mintavételezett felvételek lényegkiemeléséhez 39 dimenziós, delta és delta-delta értékkel kiegészített mel-frekvenciás kepsztrális komponenseken alapuló jellemzővektorokat hoztunk létre, és ún. vak csatornakiegyenlítő eljárást [8] is alkalmaztunk. A súlyozott véges állapotú átalakítókra (**WFST** – Weighted Finite State Transducer) [9] épülő felismerő hálózatok generálását és optimalizálását az Mtool keretrendszer programjaival végeztük, míg a tesztelés során alkalmazott egyutas mintaillesztéshez a VOXerver [3] nevű WFST dekódert használtuk. A felismerő rendszerek teljesítményének értékeléséhez szóhibaarányt (**WER** – Word Error Rate) számoltunk.

2.3 Kezdeti kísérleti eredmények

A teszteléshez összesen 21 perc felolvasott e-mailt használtunk. A felolvasott levelek mind egyetlen feladótól származnak. Ettől a feladótól egyetlen levelet sem tartalmaz a kezdeti rendszer tanítószövege. A kiértékelési eredményeket az **1. táblázatban** foglaltuk össze. A táblázatban található OOV (Out of Vocabulary) arány rövidítés a szótáron kívüli szavak tesztszövegben mutatott arányára utal.

1. táblázat: A kezdeti felismerő kiértékelési eredményei.

	Szótárméret [ezer szó]	OOV [%]	Perplexitás [-]	WER [%]
Kezdeti rendszer	263	5,0	585	**38,9**

3 Az e-mail felismerő továbbfejlesztése

Cikkünk harmadik fejezetében a kezdeti e-mail felismerő továbbfejlesztésével kapcsolatos lehetőségeket vizsgáljuk meg és értékeljük ki. Célunk az, hogy a diktálást segítő funkciókat egy olyan rendszerbe tudjuk beépíteni, mely jó kompromisszumot képvisel a felismerési pontosság és a komplexitás között.

3.1 A tanítószöveg ékezetesítése

A magyar abc számos ékezetes betűt tartalmaz, melyeket sajnos a nem vagy nem helyesen lokalizált alkalmazásokban nem tudunk bevinni. Másrészt sok felhasználó – így kollégáink közül is többen – a gyors gépelés érdekében az ékezetes betűket ékezet nélküli megfelelőjükkel helyettesíti. Az esetek döntő többségében ez az érthetőséget nem befolyásolja, sőt legtöbbször észre sem vesszük, ha ékezetek nélküli szöveget olvasunk. A felismerő rendszer azonban nem rendelkezik valódi nyelvi intelligenciával, így nyelvi modelljében nem tudja megfeleltetni egymásnak egy szó ékezetes és ékezet nélküli alakját, melynek következtében ugyanazon szókapcsolatot több különböző alakban is modellezzük. Ez rontja a statisztikai becslés pontosságát.

Megoldásként a tanítószöveg **ékezetesítése** mellett döntöttünk. Az ékezet nélküli szóalakok ékezetes változatának megkereséséhez egy speciális szótárat alkalmaztunk, melyet tanszéki kollégáink bocsátottak rendelkezésünkre [10]. Ez a szótár a leggyakoribb ékezetes párjával rendeli össze az ékezet nélküli szóalakokat. Helyzetünket nehezítette, hogy ékezetes és ékezet nélküli tanítószöveg vegyesen állt rendelkezésünkre, így a mindkét alakban értelmes szavakat valahogyan kezelnünk kellett. Kísérleti rendszerünkben azt az egyszerű megoldást követtük, hogy minden ékezet nélküli szóalakot ékezetesítettünk, ha szerepelt a szótárban. Az ékezetesített tanítószöveggel kapott eredményeket a **2. táblázatban** foglaltuk össze. Mint látható, a szótárméret csökkent, hála a kétféle formában létező szóalakok kiszűrésének. Egyedül az OOV arány romlott feltehetően a hibásan ékezetesített szavak miatt, azonban ezt a mért perplexitáscsökkenés kompenzálja, így összességében 2%-os relatív hibacsökkenést sikerült elérnünk.

2. táblázat: Az ékezetesített felismerő kiértékelési eredményei.

	Szótárméret [ezer szó]	OOV [%]	Perplexitás [-]	WER [%]
Ékezetesített rendszer	244	5,4	532	**38,1**

3.2 A rendszer személyre szabása

A hatékony diktálórendszerek használatba vételét mindig egy tanítási vagy adatgyűjtési feladat előzi meg, ezért úgy döntöttünk, hogy mi is felhasználunk beszélőspecifikus adatokat a rendszerünk optimalizálásához. Első lépésben a diktálórendszer nyelvi modelljét egészítettük ki a tesztanyaghoz tartozó feladó korábbi leveleivel. Ezt az összesen 83 ezer szót tartalmazó tanítószöveget nyelvimodell-interpolációs technika segítségével egyesítettük az ékezetesített kezdeti rendszer modelljével. Az interpolált nyelvi modellek készítéséhez és optimalizálásához az SRILM beépített lineáris interpolációs és perplexitásszámító eljárásait használtuk. Az új nyelvi modellel kapott eredményeket a **3. táblázatban** mutatjuk be.

A szöveges adatok mellett az adott beszélőtől származó hanganyagok is felhasználhatóak a rendszer személyre szabása során. A kézi munka minimalizálása érdekében a rögzített tesztanyagon **felügyelet nélküli adaptációt** hajtottunk végre. Az adaptált akusztikus modellel végzett teszt eredményét szintén a 3. táblázat tartalmazza.

Mint az a táblázatból is kiolvasható, a nyelvi modell adaptációval az ékezetesített rendszerhez képest 3%-os relatív szóhiba-arány csökkenés érhető el. Ezen felül azonban további 30%-os javulást mértünk az akusztikus modell adaptálásával. Ez alapján elmondható, hogy a kezdeti nyelvi modell távolról sem állt olyan messze az optimálistól, mint a kiindulás EKR akusztikus modell, mely teljes mértékben a feladattól független adatokon került betanításra.

3. táblázat: A személyre szabott felismerő kiértékelési eredményei.

	Szótárméret [ezer szó]	OOV [%]	Perplexitás [-]	WER [%]
Nyelvimodell-adaptált rendszer	246	5,0	501	**37,0**
+Akusztikusmodell-adaptáció				**26,0**

4 Kiegészítő funkciók a diktáláshoz

A korábban fejlesztett felismerőrendszereinkben a beszédet mint szótári szavak sorozatát modelleztük. A közelmúltban azonban eredményesen teszteltünk egy újabb megközelítést, melyben a szavak mellett más, a spontán beszédre jellemző hangesemányeket is modelleztünk [11]. Ehhez hasonlóan a diktálási feladat során felmerülő írásjeleket és speciális szimbólumokat is modelleznünk kell, ha hatékonyan szeretnénk őket a felismerőbe integrálni. A problémát érdemes két részre osztani. Egyrészt a kiegészítő funkciót ellátó új szótári elemeket be kell építeni a nyelvi modellbe, másrészt gondoskodni kell az akusztikai szintű modellezésükről is.

4.1 Nyelvi modell felkészítése a diktálási feladatra

A nyelvi modell struktúrájának megváltoztatásához az e-mail felismerő tanítószövegén kell változtatásokat végezni. Elsősorban azt kellett eldönteni, hogy pontosan milyen elemeket is szeretnénk modellezni, és ennek megfelelően kellett átalakítani a tanítókorpusz normalizálását. A kiválasztás során arra törekedtünk, hogy a bevezetett új lexikai elemek segítségével az egyszerűbb elektronikus levelek további kézi kiegészítés nélkül is bevihetőek legyenek. Mint az a **4. táblázatból** is kiolvasható, a legalapvetőbb írásjelek és az „új sor” parancs mellett beépítettünk két emotikont is a nyelvi modellbe, mert úgy ítéltük meg, hogy ezek használata nagyon elterjedt.

4. táblázat: Diktálási szimbólumok a nyelvi modellben.

Felszíni forma	**.**	**!**	**?**	**,**	**\n**	**:)**	**:(**
Nyelvimodell-szimbólum	<pont>	<fj>	<kj>	<vessző>	<nl>	<mosoly>	<szomorkodás>

Minta a diktáláshoz előkészített tanítószöveg egy sorára:

„a Redmine-on keresetem a VOXerver dokumentációját <vessző> de végül nem találtam meg <pont> <nl>”

4.2 A diktálási szimbólumok modellezése

4.2.1 Hagyományos megközelítés

A 4. táblázatban bemutatott új szimbólumok akusztikai szintű modellezésére a legelterjedtebben használt megoldás, hogy egy meghatározott hangsorozatra képezzük le őket. A mi rendszerünkben beépített leképezéseket az **5. táblázatban** foglaltuk össze. Nyilvánvaló előnye a megközelítésnek, hogy nagy pontossággal lehet ilyen módon a diktálási szimbólumokat detektálni, amit ki is használ a legtöbb ma forgalomban lévő automatikus diktálórendszer. Nem mehetünk el azonban szó nélkül a hátrányai mellett sem. A diktálás során kényelmetlenséget jelent, hogy minden írásjelet ki kell ejtenünk. A felhasználók számára ez egyáltalán nem természetes, hiszen így a rendszer használata gyakorlást igényel, sőt véleményünk szerint egyes felhasználókat pont ez a fajta kényelmetlenség tart távol a diktálórendszerek használatától.

5. táblázat: Diktálási szimbólumok a nyelvi modellben.

Felszíni forma	**.**	**!**	**?**	**,**	**\n**	**:)**	**:(**
Kiejtett alak	p-o-n-t	f-e-l -k-i-á-l-t-ó -j-e-l	k-é-r-d-ő- j-e-l	v-e-sz-ő	ú-j-s-o-r	m-o-s-o-j	Sz-o-m-o-r -k-o-d-á-s

4.2.2 Prediktív megközelítés

A problémát jobban megvizsgálva észrevehetjük, hogy vannak olyan írásjelek, melyeket önmagában a nyelvi modell képes lehet hatékonyan előre jelezni anélkül, hogy kiejtett alakjukat be kellene diktálni. Ilyen lehet, a „vessző”, mondatzáró „pont” és bizonyos esetekben a „kérdő- és felkiáltójelek”. Az „új sor” parancs és az emotikonok használata sokkal kevésbé szabályokhoz kötött, így ezek detektálása csak a hagyományos módszerrel képzelhető el hatékonyan. Kísérleti rendszerünkben azonban az összes diktálási szimbólumot megkíséreljük a nyelvi modellre támaszkodva detektálni, melynek érdekében akusztikai szinten az összeset semmi vagy szünet (**sp**) modellre képezzük le.

4.3 Kísérleti eredmények

A fejezetben található eredmények a 3.2-es pontban kapott rendszer továbbfejlesztésével jöttek létre.

4.3.1 Hagyományos megközelítés

A hagyományos megközelítés kiértékeléséhez felhasznált tesztfelvételekben az 5. táblázatban bemutatott összes szimbólum bemondásra került kiejtett alakjuknak megfelelő formában. Kísérleteink várakozásainknak megfelelően azt mutatták, hogy ezzel a megközelítéssel a diktálási szimbólumok közel tökéletes pontossággal felismerhető-

ek, miközben a normál szavakra számított hiba sem növekedett meg szignifikáns mértékben. A helyesen felismert szimbólumok aránya (**Corr.** – Correct Rate) átlagosan **93,1%**-os volt.

4.3.2 Prediktív megközelítés

Prediktív megközelítésünk tesztelésének célja elsősorban az volt, hogy kiderítsük, mely diktáláskor fontos lexikai elem felismerését érdemes a nyelvi modellre bízni, és így egyszerűsíteni a diktálást. Tesztelési célokra itt a felvételek egy olyan változatát használtuk, melyben semmiféle diktálási szimbólum nem jelenik meg kiejtett formájában. A kapott eredményeket a **6. táblázat**ban mutatjuk be. A táblázatban csak a „vessző" és „pont" szimbólumok eredményeit tüntettük fel, ugyanis a többi szimbólumra nem kaptunk értékelhető eredményt. A helyesen felismert szimbólumok aránya a „vessző" esetén majdnem 73%-os, azaz a vesszők közel háromnegyedét képes helyesen detektálni a prediktív rendszer. A nem elhanyagolható mértékű beszúrási hiba figyelembevételével is azt mondhatjuk, hogy az automatikus „vessző" detekció beépítése megfontolandó végső rendszerünkbe. A „pont" esetében ugyanez már nem mondható el. Mindössze minden tizedik mondatvégi pontot sikerült helyesen beilleszteni, ami egyelőre nem teszi lehetővé ennek a funkciónak a használatát. Mindezek mellett jó hír, hogy a diktálási szimbólumok beépítése csak minimális hatással volt a többi szó felismerési hibájára. A 3.2-es pontban ismertetett rendszerhez képest mért kevesebb mint 3%-os relatív hibaarány csökkenés elhanyagolhatónak tekinthető.

6. táblázat: A prediktív megoldással kiegészített felismerő kiértékelési eredményei.

	<vessző>		<pont>		WER* [%]
	Corr. [%]	WER [%]	Corr. [%]	WER [%]	
Prediktív megközelítés	72,9	58,9	10,8	92,3	**26,7**

*A diktálási szimbólumok kivételével az összes szón számolt szóhiba-arány

5 Összefoglalás

Cikkünkben bemutattuk egy olyan, magyar nyelven egyedülálló diktálórendszer fejlesztésének lépéseit, mely akár a felhasználó eszközén futva, egyes írásjelek automatikus elhelyezése mellett képes számok, emotikonok, nagybetűs szavak és rövidítések felismerésére is. Első lépésben a kezdeti rendszerünket ismertettük, melynek hibaarányát a tanítószöveg ékezetesítésével és a modellek személyre szabásával 33%-kal sikerült csökkenteni. Ezután a diktáláshoz szükséges kiegészítő elemek beépítési lehetőségeit vizsgáltuk meg. A legfontosabb írásjelek mellett a soremelés funkciót és a két leggyakrabban használt emotikon felismerését is lehetővé tettük rendszerünkben. Kísérleteink alapján elmondható, hogyha a hagyományos megközelítést követve parancsszavakat rendelünk ezekhez az elemekhez, a detekciójuk minimális felismerési hiba mellett biztosítható. Hátrányként jelentkezik azonban az állandó bemondásukkal járó kényelmetlenség. Ennek kivédése érdekében kísérletet tettünk a diktálást segítő lexikai elemek automatikus észlelésére. Eddigi eredményeink azt mutatják,

hogy a nyelvi modell alapján csak a „vessző” kiváltására kapunk megfelelően pontos előrejelzést, ami érthetővé teszi, miért nem jelentek meg még effajta megoldások az ipari rendszerekben. Véleményünk szerint igény ugyanakkor lenne rá, így ez továbbra is érdekes kutatási terület marad.

További vizsgálataink középpontjában a prediktív írásjel-detekciót helyezzük. Meg kívánjuk vizsgálni, hogy a „vessző” automatikus elhelyezésekor keletkező hiba a gyakorlatban mennyire tolerálható, illetve lehetővé kívánjuk tenni, hogy az automatikus beszúrás mellett normál bemondással is elhelyezhessünk vesszőt. Ezen kívül további kényelmi funkcióként a köszönési és az aláírás formátum személyre szabhatóságát is meg szeretnénk oldani.

Köszönetnyilvánítás

Kutatásunkat a Mindroom (KMOP-1.1.3-08/A-2009-0006), Paelife (AAL-08-1-2011-0001) és a BelAmi (OMFB-00736/2005 BELAMI_H) projektek támogatták.

Hivatkozások

1. Szarvas, M., Furui, S.: Evaluation of the stochastic morphosyntactic language model on a one million word Hungarian task. In: EUROSPEECH2003 (2003) 2297–2300
2. Vicsi, K., Velkei, S., Szaszák, Gy., Borostyán, G., Teleki, C., Tóth, S. L., Gordos, G.: Középszótáras, folyamatos beszédfelismerő rendszer fejlesztési tapasztalatai. In: II. Magyar Számítógépes Nyelvészeti Konferencia (2005) 348–359
3. Tarján, B., Mihajlik, P., Balog, A., Fegyó, T.: Evaluation of lexical models for Hungarian Broadcast speech transcription and spoken term detection. In: 2nd International Conference on Cognitive Infocommunications (CogInfoCom) (2011) 1–5
4. Trón, V., Gyepesi, Gy., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: Open Source Word Analysis. In: Proc. of the ACL Workshop on Software (2005) 77–85
5. Chen, S. F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Computer Speech & Language, Vol. 13, No. 4 (1999) 359–393
6. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing (2002) 901–904
7. Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P. C.: The {HTK} Book. Version 3.4. Cambridge, UK: Cambridge University Engineering Department (2006)
8. Mauuary, L.: Blind equalization for robust telephone based speech recognition. In: Proc. of the European Signal Processing Conference (1996) 359–363
9. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech and Language, Vol. 16, No. 1 (2002) 69–88
10. Zainkó, Cs., Csapó, T. G., Németh, G.: Special speech synthesis for social network websites. In: Lecture Notes in Computer Science 6231 (2010) 455–463
11. Sárosi, G., Tarján, B., Balog, A., Mozsolics, T., Mihajlik, P., Fegyó, T.: On Modeling Non-word Events in Large Vocabulary Continuous Speech Recognition. In: 3rd International Conference on Cognitive Infocommunications (CogInfoCom) (2012) 649–653

Hogyan tanuljunk kevés információból is? A RIP-algoritmus továbbfejlesztett változatai

Biró Tamás

Amszterdami Egyetem (UvA)
Spuistraat 210, Amszterdam, Hollandia, e-mail: birot@nytud.hu

Kivonat A nyelvtanuló gyakran nem fér hozzá olyan információhoz, amely a nyelvészeti elméletekben központi szerepet játszik. Ez az információhiány a számítógépes szimulációk szerint hátráltathatja a nyelvelsajátítást. Kutatásom során az OT tanulóalgoritmusok sikerességét javítom Prince és Smolensky RIP-eljárásának továbbfejlesztésével.[1]

Kulcsszavak: Optimalitáselmélet (OT), Robust Interpretive Parsing, szimulált hőkezelés/lehűtés, genetikai algoritmusok, tanulóalgoritmusok.

1. Bevezetés: hiányzó információ a tanulás során

Vajon a *John loves Mary* mondat egy SVO vagy egy OVS nyelvből származik? Helyezzük magunkat a nyelvtanuló helyébe, aki hallja ezt a nyelvi adatot, és megfelelő ismerettel is rendelkezik a világról (vagyis tud a két személy közötti kölcsönös szerelemről): vajon milyen következtetést vonjon le a nyelvtanuló az elsajátítandó célnyelv szórendjére vonatkozóan? Amennyiben ezen a ponton (helytelenül) tárgy-ige-alany szórendet feltételez, akkor ez a nyelvi adat megerősítheti a nyelvtanulót téves hipotézisében, és a tanulási folyamat félrecsúszhat. Ha azonban egy más, óvatosabb algoritmust követ, és számol azzal, hogy jelenlegi hipotézise akár hibás is lehet, miközben a nyelvi adat több módon interpretálható, akkor a tanulás sikerrel járhat – mint azt rövidesen bemutatom.

A mondattanban az alany és a tárgy megkülönböztetése központi szerepet játszik, de az angol nyelvet éppen elsajátító nyelvtanuló számára nem hozzáférhető információ az, hogy az informáns (tanító) mely főnévi csoportot szánta alanynak, és melyiket tárgynak. A nyelvtan számos más pontján is hasonló problémák merülnek fel. Tizenegy hónapos kislányom megsimogatott a [*Mutasd meg, hol van*] *apa szeme!* utasításra, mert még nem sajátította el a [s]∼[š], valamint az [e]∼[i] közötti fonológiai különbségeket. Ezért a *szeme*∼*simi* párt szabad alternációként, nem pedig minimálpárként értelmezte. Apaként bízom benne, hogy kislányom esetében ez az egyszeri eset nem tereli vakvágányra a magyar fonológia elsajátítását.

[1] A szerző köszönetét fejezi ki a *Holland Tudományos Kutatási Alapnak* (NWO), amely a 275-89-004 számú Veni-projekt keretében az ismertetett kutatást támogatta.

Számítógépes nyelvészként célom a meglévő tanulóalgoritmusok továbbfejlesztése ugyanezen problémák elkerülése végett. Kutatásom tárgya az egyik leggazdagabb tanulhatósági irodalommal rendelkező kortárs nyelvészeti keret, az *Optimalitáselmélet* (OT) [1]. Az előbbiekben bemutatott problémára az OT hagyományos megoldása a *Robusztus Interpretatív Parszolás* (RIP) [2], amelyet a 3. fejezetben tárgyalok. A RIP teljesítménye azonban kívánnivalót hagy maga után. Ezért a 4. fejezetben két alternatívát mutatok be, amelyek teljesítményét az 5. fejezetben tesztelem.

Az első javaslat [3] a szimulált hőkezelés technikájából merít, és Boltzmann-eloszlást vezet be a megfigyelt nyelvi adat lehetséges interpretációin. A második javaslatot [4] a genetikai algoritmusok ihlették: párhuzamosan több, független tanulóalgoritmus fut, amelyek közösen interpretálják a bejövő nyelvi adatokat. Mielőtt azonban ezekre rátérnénk, foglaljuk össze az OT-val és tanulóalgoritmusaival kapcsolatos tudnivalókat.

2. Az optimalitáselmélet és tanulóalgoritmusai

Az *optimalitáselmélet* (*Optimality Theory*, OT) [1] alapgondolata az, hogy egy u bemenet (például mögöttes reprezentáció) arra a kimenetre (felszíni reprezentációra) képeződik le, amely optimalizál egy célfüggvényt. A gondolat önmagában nem új, hiszen számos tudományterület a fizikától a közgazdaságtanig – közöttük sok számítógépes kognitív modell is – célfüggvények optimalizációjával magyarázza jelenségeit. A nyelvészetben is gyakran hivatkozunk a „minél jobb" alakra. A nyolcvanas években a generatív nyelvészetben (különösen a fonológiában) megnőtt a teleológikus érvelés szerepe: az újraíró szabályok *célja* az, hogy valamilyen elveknek megfeleljen – vagy „jobban" megfeleljen – a nyelvtani alak. Az optimalitáselmélet ezeket a nyelvészeti trendeket formalizálja, és így a formális OT a *számítógépes elméleti nyelvészet* egyik legdinamikusabban fejlődő ága lett.

Hasonlóan a nyelvészeten kívüli – például fizikai, közgazdaságtani vagy pszichológiai – optimalizációs modellekhez, valamint közeli rokonához, a *harmónia-nyelvtanhoz* is [5], az OT különböző szempontokat (*constraints*, magyarul *megszorítások* vagy *korlátok*, vö. [6]) „gyúr össze" egyetlen célfüggvénnyé. Ezek a megszorítások gyakran egymással összeegyeztethetetlen és összemérhetetlen elvárásokat támasztanak a grammatikus alakkal szemben. A chomskyánus felfogással ellentétben, a grammatikus alakok megsérthetnek egyes megszorításokat, azonban a cél az, hogy „összességben minél jobban teljesítsenek".

Formálisan megfogalmazva: Egy u bemenetet (mögöttes alakot) a Gen *generátorfüggvény* a *jelöltek* (*candidates*: potenciális felszíni alakok) $\mathrm{Gen}(u)$ halmazára képezi le. Majd az optimalitáselmélet alapaxiómája azt mondja ki, hogy az u bemenethez tartozó $\mathrm{SF}(u)$ grammatikus felszíni alak optimalizálja a $H(c)$ célfüggvényt, a *Harmóniafüggvényt*:

$$\mathrm{SF}(u) = \underset{c \in \mathrm{Gen}(u)}{\arg\,\mathrm{opt}}\, H(c) \tag{1}$$

Az optimalitáselmélet a nyelvek (nyelvtípusok) közötti különbségeket eltérő célfüggvényekkel modellezi, melyeket más és más jelöltek optimalizálnak. Hogy az optimalizálás mit is jelent – maximalizálást vagy minimalizálást –, attól függ, hogy hogyan reprezentáljuk a célfüggvényt. Hagyományosan a $H(c)$ harmónia maximalizálásáról szokás beszélni. De az alábbiakban mi inkább megspórolunk magunknak egy negatív előjelet: a megszorítások sértéseinek a minimalizálása, és így a megszorításokból összerakott célfüggvény minimalizálása lesz a célunk.

Ha az egyes C_i megszorításokat a constraintek Con univerzális halmazából vett valós értékű függvényeknek tekintjük,[2] akkor ezek lineáris kombinációja egy valósértékű célfüggvényt eredményez:

$$H(c) = \sum_{i=0}^{n-1} g_i \cdot C_i(c) \tag{2}$$

Ezt nevezzük harmónianyelvtannak, és itt az (1)-beli optimum egyszerűen a valós számok halmazán vett minimumot jelenti. A lineáris kombináció g_i súlyai határozzák meg azt, hogy melyik megszorítás milyen eréllyel szól bele a grammatikus alak meghatározásába. A legtöbb nyelvészeten kívüli modell (például a közgazdaságtudományban és a kognitív tudományokban) hasonló optimalizációs elveket követ.

Ezzel ellentétben, az optimalitáselmélet nem valósértékű függvénnyé „gyúrja össze" a megszorításokat, hanem egy *hierarchiába* rangsorolja őket. A magasabbra rangsorolt megszorítás perdöntő: ha azt egy jelölt más jelölteknél súlyosabban sérti meg, akkor végképp elbukik, hiába viselkedik amúgy kitűnően az alacsonyabbra rendezett megszorítások szempontjából. Az ezen elvet (*szigorú dominancia*, *strict domination*) teljesítő harmóniafüggvényt többféle módon is reprezentálhatjuk: megszorításértések csomagjaként (multihalmazaként) [1], polinomokként vagy halmazelméleti rendszámokként [7]. A legegyszerűbb a vektorként történő reprezentáció, amelyeket lexikografikusan rendezhetünk az optimalizálás során:[3]

$$H(c) = \big(C_{n-1}(c), \ldots, C_1(c), C_0(c)\big) \tag{3}$$

A constraintek indexe tükrözi a rangsorolásukat: $C_{n-1} \gg \ldots \gg C_1 \gg C_0$. A c jelölthöz rendelt $H(c)$ vektor $n-i$-ik komponense a C_i megszorításnak felel meg, jelentése pedig az, hogy milyen mértékben (a legtöbb nyelvészeti modellben: hányszor) sérti meg a c jelölt a C_i megszorítást. A $H(c)$ vektor nem más, mint c sora az ismert OT-táblázatban, a csillagokat azok számával helyettesítve.

[2] Az optimalitáselmélet matematikailag helyes definíciójához azt is feltételeznünk kell, hogy az egyes megszorítások értékkészlete egy-egy jólrendezett halmaz [7]. A nyelvészeti gyakorlatban ez teljesül, hiszen a megszorítások általában nemnegatív egész értéket vesznek fel: nullát, ha a jelölt megfelel a megszorításbeli követelménynek, vagy egy pozitív egész számot, ha valahányszorosan megsérti azt.

[3] Lásd például [8]-t. [9, p. 1009] körbeírja a vektorreprezentációt, de nem nevezi néven. Tudtommal [10] hivatkozik először vektorokra, míg [11] a lexikografikus rendezésre. A két kifejezés [12]-ben találkozik először egymással.

Ha $H(c_1)$ lexikografikusan kisebb $H(c_2)$-nél, akkor c_1 harmonikusabb c_2-nél. Nevezzük *fatális megszorításnak* azt a C_f megszorítást, amelyre $C_f(c_1) \neq C_f(c_2)$, de minden magasabbra rendezett megszorítás azonosan értékeli ezt a két jelöltet. A fatális megszorítás felel meg a $H(c_1) - H(c_2)$ különbségvektor első nem-nulla elemének. Ez az elem határozza meg $H(c_1)$ és $H(c_2)$ lexikografikus rendezését: $C_f(c_1) < C_f(c_2)$ akkor és csak akkor, ha $H(c_1)$ lexikografikusan kisebb, mint $H(c_2)$. Átfogalmazva olyan formába, ahogy azt rövidesen használni fogjuk: ha c_1 harmonikusabb, mint c_2, akkor a fatális megszorítás c_1-et preferálja.

Mivel a H harmóniafüggvény értékkészlete n jólrendezett halmaz Descartes-szorzata, ezért maga az értékkészlet is jólrendezett halmaz a lexikografikus rendezés mellett. Következésképpen, valóban jól definiált az OT alapaxiómája:

$$\mathrm{SF}(u) = \underset{c \in \mathrm{Gen}(u)}{\arg\,\mathrm{opt}}\, H(c) \tag{4}$$

azaz az u bemenethez (mögöttes reprezentációhoz) tartozó $\mathrm{SF}(u)$ grammatikus felszíni reprezentáció optimalizálja a harmóniafüggvényt. Elvileg lehetséges, hogy két felszíni reprezentáció ugyanúgy sértse valamennyi megszorítást, és egyaránt optimalizálják a harmóniafüggvényt: ebben az extrém esetben az OT mindkét alakot grammatikusnak jósolja. A (4) egyenlőségben az optimalizálás lexikografikus minimalizálást jelent a fenti gondolatmenetünk értelmében. Azonban a szakirodalom, egy negatív előjelet helyezve $H(c)$ elé, a harmóniafüggvény maximalizálásáról beszél. E két megközelítés között nincs érdemi különbség.

Az optimalitáselmélet fősodra szerint mind a Gen függvény, mind a Con halmaz univerzális. A nyelvtanok közötti eltérést kizárólag a Con-beli megszorítások *rangsorolása* okozza. Két természetes nyelv nyelvtana a harmóniafüggvényükben különbözik egymástól, mégpedig abban, hogy a (3)-beli vektor komponenseit hogyan permutálják.

Optimalitáselméleti keretben a *tanuló algoritmus* feladata tehát a következő: adott (u_k, s_k) bemenet–kimenet párokhoz megtalálni azt a H függvényt, a komponensek azon permutációját, a megszorítások azon rangsorolását, amely mellett minden k-ra teljesül $s_k = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H(c)$. Az *offline algoritmusokban*, mint amilyen [13] *Recursive Constraint Demotion* algoritmusa, a tanítóadatokat, a mögöttes alak–felszíni alak párokat, egyszerre kapja meg a tanuló, mielőtt ezekből kikövetkeztetné a célnyelvtant. Ezek az algoritmusok azonban nyelvelsajátítási modellként kevéssé plauzibilisek. Így fordítsuk a figyelmünket inkább az *online algoritmusokra*, amelyek az adatokat folyamatosan adagolják a nyelvtanulónak.

Ez utóbbiak *hibavezérelt* (*error-driven*) megközelítések. A tanuló egy $H^{(0)}$ nyelvtannal (harmóniafüggvénnyel, megszorítás-rangsorolással) indul, amelyet fokozatosan módosít a megfigyelései függvényében. $H^{(0)}$ lehet egy véletlen hierarchia, vagy valamely „veleszületettnek” gondolt rangsorolás. Például a gyermeknyelvi adatok alapján szokás amellett érvelni, hogy kezdetben a jelöltségi (*markedness*) megszorítások magasabbra vannak rendezve, mint a hűségi (*faithfulness*) megszorítások. A tanulás egy pontján a tanuló által feltételezett $H^{(k-1)}$ nyelvtan predikciója az u_k-hoz tartozó jelöltre: $l = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H^{(k-1)}(c)$.

Ha ez az l (*loser form* a szakirodalomban) megegyezik a megfigyelt s_k-val (az alábbiakban w, mint *winner form*), akkor tanulónk örül a sikernek, és reménykedik, hogy elsajátította a célnyelvtant, minden más bemenetre is eltalálná a kimenetet. Amennyiben azonban l különbözik s_k-tól, a tanuló annak örül, hogy lehetősége van tanulásra: igyekszik úgy módosítani a nyelvtanát, hogy legközelebb $H^{(k)}$ már a helyes alakot jósolja. De legalábbis egy olyan nyelvtan felé közelítsen, amely a helyes w (azaz s_k) alakokat produkálja. A sikeres tanulás végén H^{∞} megegyezik a tanító H_t nyelvtanával, vagy legalább ekvivalens vele: minden (megfigyelhető) bemenetre azonos kimenetet jósol.

Hogyan módosítja a tanuló a nyelvtanát, amikor hibát észlel? Egyes megszorításokat feljebb, másokat lejjebb rangsorol annak érdekében, hogy közelebb kerüljön a célnyelvtanhoz. A tanító H_t nyelvtana, a célnyelvtan, az u_k mögöttes alakhoz a $w = s_k = \arg\,\mathrm{opt}_{c \in \mathrm{Gen}(u_k)} H_t(c)$ jelöltet rendeli. Mit jelent az, hogy l különbözik w-től? Azt, hogy H_t szerint w harmonikusabb l-nél, de $H^{(k-1)}$ szerint l harmonikusabb w-nél. Tehát, mint fentebb láttuk, a H_t-beli fatális megszorítás w-t kedveli, míg a $H^{(k-1)}$-beli fatális megszorítás l-t. A tanuló ebből azt a következtetést vonja le, hogy valamelyik w-t kedvelő megszorítást az l-t kedvelő megszorítások fölé kell rendeznie. Ezért az online OT tanulóalgoritmusok végigtekintik a Con-beli megszorításokat. Az l-t kedvelőket (vagy azok egy részét) lejjebb rendezik, a w-t kedvelőket pedig (esetleg) feljebb. Hogy pontosan hogyan teszik ezt, abban eltérnek egymástól a különböző algoritmusok [14,2,15,16,17,18].

3. Amikor a tanuló nem kap meg minden információt

Eddig feltételeztük, hogy a tanuló számára világos, melyik w jelölttel kell összevetnie az aktuális nyelvtana által generált l jelöltet. Ez azonban nincs mindig így, amint azt a bevezető fejezetben már láttuk. A megfigyelt nyelvi adat (*overt form*) nem feltétlenül jelölt OT értelemben (*candidate*). Utóbbi tartalmazhat olyan nyelvtani információt (például a szintaktikai frázisok és a fonológiai lábak határait jelző zárójeleket), amelyek az előbbiből hiányoznak. A hallható nyelvi adat nem feltétlenül felel meg egyetlen w jelöltnek, hanem jelöltek egy tágabb W halmazára képezhető csak le (például az azonos lineáris szerkezetet leíró fák erdejére). A W-beli jelöltek azonban egymástól eltérő módon sértik az egyes megszorításokat, és így a tanuló számára kérdéses marad, hogy mely megszorítást kell lejjebb, melyeket pedig feljebb rangsorolnia.

Egy korábbi kutatásban például a tagadó mondatok tipológiáját és történeti fejlődését vizsgáltuk [19]. A tagadószó (SN) megelőzheti az igét (SN V szórend, mint a magyarban, az olaszban és az ófranciában), követheti azt (V SN, mint a törökben vagy az élőnyelvi franciában), és körbe is veheti (SN V SN, mint az irodalmi franciában és az óangolban). Az utóbbi szórend azonban két különböző fastruktúrának is megfelelhet: [SN [V SN]] vagy [[SN V] SN]. A frázishatárok a szintaktikai elméleteknek szerves részei, de nem hallhatóak, nincsenek jelen a nyelvtanuló számára hozzáférhető nyelvi adatban. Az a nyelvtanuló gyermek, aki azt figyeli meg, hogy a célnyelv két részből álló tagadószerkezetet tartal-

maz (SN V SN), vajon miből fog rájönni, hogy a fenti két jelölt közül melyik grammatikus jövendőbeli anyanyelvében?

Tekintsük a következő (leegyszerűsített) példát. A Gen függvény a következő három jelöltet generálja (vagy a többi jelöltet már más megszorítások kiszűrték): [SN V], [[SN V] SN] és [SN [V SN]]. Három megszorításunk közül a *NEG minden egyes SN tagadószót egy megszorítássértéssel bünteti. A V-RIGHT és a V-LEFT megszorítások pedig a V-t közvetlenül tartalmazó frázis (mondjuk V' vagy VP) szerkezetére vonatkoznak: akkor teljesülnek, ha a V ennek a frázisnak a jobboldali, ill. baloldali eleme. Tehát a következő OT-táblázatot kapjuk:

	Tanuló →		← Tanító	
		*NEG	V-RIGHT	V-LEFT
l	[SN V]	1	0	1
w	[[SN V] SN]	2	0	1
	[SN [V SN]]	2	1	0

(5)

Képzeljük el, hogy a célnyelvtan V-LEFT ≫ V-RIGHT ≫ *NEG, vagyis a tanító (informáns) jobbról balra olvassa a fenti táblázatot. Számára az [SN [V SN]] alak a grammatikus, ami SN V SN-ként hangzik. Tegyük fel azt is, hogy a tanuló, pechjére, éppen az ellenkező hierarchiát feltételezi, a fenti táblázatot balról jobbra olvassa: *NEG ≫ V-RIGHT ≫ V-LEFT. Ő, ha rajta múlna, [SN V]-t mondana, de ez az l forma másként hangzik. Amint hallja a tanító által produkált alakot, észleli az eltérést, és beindul a hibavezérelt online tanuló algoritmusa. A nyelvtanát úgy szeretné módosítani, hogy SN V helyett legközelebb SN V SN-t mondjon. Azaz a nyelvtana egy másik jelöltet hozzon ki optimálisnak... Jó, de melyiket? [[SN V] SN]-t vagy [SN [V SN]]?

Tesar és Smolensky [14,2] azt javasolták, hogy a tanuló használja a saját nyelvtanát arra, hogy kiválassza az SN V SN két lehetséges értelmezése közül azt a w alakot, amellyel össze fogja vetni a saját maga által produkált l alakot. A tanuló nyelvtana felől (balról jobbra) nézve a táblázatot látjuk, hogy ő az [[SN V] SN] jelöltet jobbnak találja, mint az [SN [V SN]] jelöltet. Vagyis arra fog törekedni, hogy l helyett w-t hozza ki legközelebb optimálisnak. Több online OT tanulóalgoritmus létezik, amelyek részleteikben különböznek egymástól, de az alapgondolatuk azonos: ha egy megszorítás l-t jobbnak találja, mint w-t, akkor lejjebb kell rendezni (legalábbis, ha magasra volt eredetileg rangsorolva), ha pedig w-t találja jobbnak l-nél, akkor (bizonyos algoritmusban) feljebb.

Esetünkben egyetlen megszorítás van, amelyik eltérően értékeli l-t és w-t: a *NEG megszorítás l-t preferálja, és ezért lejjebb kell rangsorolni. A tanuló így eljuthat a V-RIGHT ≫ *NEG ≫ V-LEFT, majd a V-RIGHT ≫ V-LEFT ≫ *NEG hierarchiákhoz. Azonban, figyeljük meg, a tanuló mindvégig az [SN V] jelöltet fogja grammatikusnak tartani, a megfigyelt SN V SN alakot pedig mindig [[SN V] SN]-ként fogja értelmezni. Előbb-utóbb *NEG a rangsorolás aljára, a tanuló pedig patthelyzetbe kerül: az algoritmus elakad, az egyetlen átrangsorolandó megszorítást nincs már hova tovább átrangsorolni. A gondot az okozza, hogy a megoldás V-LEFT és V-RIGHT rangsorolásának a felcserélése lenne, de erre az algoritmus „nem jön rá". Mindvégig, amíg ez a csere nem történik meg, a

tanuló [SN V]-t tekinti l-nek és [[SN V] SN]-t w-nek, utóbbi produkálására törekszik. Ekkor valójában lehetetlent tűz ki célul: az [[SN V] SN] jelölt harmonikusan korlátolt (*harmonically bounded* [20]), egyetlen megszorítás szempontjából sem jobb, mint [SN V], és ezért nem létezik olyan rangsorolás, amely [[SN V] SN]-t hozná ki győztesnek. Hogyan lehet kitörni ebből a patthelyzetből?

Foglaljuk össze az eddigieket: a hibavezérelt online OT tanulóalgoritmusok (1) összehasonlítják a megfigyelt w jelöltet – vagy a megfigyelt alak egyik lehetséges w interpretációját – a tanuló által hibásan grammatikusnak vélt l jelölttel, és ha ezek egymástól eltérnek („hiba" lép fel), akkor (2) meghatározzák, hogy melyik megszorítás preferálja l-t, és melyik w-t, végül (3) előbbieket lejjebb, utóbbiakat feljebb rendezik. A *szétválasztás menetrendje*:

Minden $C_i \in \mathsf{Con}$ megszorításra,
1. ha $C_i(w) > C_i(l)$, akkor a C_i megszorítás l-t preferálja;
2. ha $C_i(w) < C_i(l)$, akkor a C_i megszorítás w-t preferálja.

Az l jelölt meghatározása, hibavezérelt algoritmusról lévén szó, természetesen a tanuló (egyelőre még) hibás nyelvtanától függ. A probléma abból származik, hogy szintén erre a hibás hierarchiára támaszkodunk w meghatározásánál, azaz a megfigyelés interpretálása során. Bár mindegyik W-beli jelölt ugyanúgy hangzik, de egyetlen w jelöltet választunk ki közülük a tanuló hibás nyelvtana segítségével. Egy rossz döntés ezen a ponton félreviheti az egész tanulási folyamatot. Milyen alapon bízzuk a tanító adatok értelmezését egy nyilvánvalóan téves hipotézisre? Tesar és Smolensky, amikor az eddigiekben leírt, *Robust Interpretive Parsing* (RIP, 'Robusztus Interpretatív Parszolás') nevű eljárásukat javasolták, az *Expectation–Maximization*-módszerek konvergenciáját látva azt remélték, hogy iteratív módon, előbb-utóbb, a tanuló eljuthat a célnyelvtanhoz. Sajnos azonban a kísérleteik azt mutatták, hogy ez nincs mindig így: néha végtelen ciklusba fut a tanuló, néha pedig – akárcsak a fenti példánkban – zsákutcába.

4. Két kiút a zsákutcából: Általánosított RIP

Figyeljük meg, hogy a szétválasztás fenti menetrendje során valójában érdektelen, hogy pontosan melyik jelöltet is választjuk w-nak. Ami számít, az w viselkedése az egyes megszorítások szempontjából. Nem szükséges rámutatnunk valamelyik jelöltre: elegendő meghatároznunk azt a határértéket, amellyel $C_i(l)$-t összehasonlítjuk. Ha $C_i(l)$ kevesebb a határértéknél, akkor a C_i megszorítás „l-et preferálja", és alacsonyabbra kell rangsorolni. Ha pedig $C_i(l)$ több, akkor C_i „w-t preferálja", és (az algoritmus részleteitől függően) magasabbra rangsorolandó. Az alábbiakban ezt a $C_i(W)$ határt az egész W halmazból számoljuk ki.

A fenti példánkban a tanuló, bár [SN V]-t mondana, de a hallott SN V SN alakról nem tudja eldönteni, hogy az hogyan interpretálandó: vajon a tanító nyelvtana szerint [[SN V] SN] vagy [SN [V SN]] a grammatikus? A maximumentrópia módszerek azt javasolják, ha nem tudunk dönteni két lehetőség közül, akkor adjunk mindkettőnek egyenlő esélyt. Tegyünk így most is, és átlagoljuk a táblázat két sorát:

		*Neg	V-right	V-left
l	[SN V]	1	0	1
w_1	[[SN V] SN]	2	0	1
w_2	[SN [V SN]]	2	1	0
W	w_1 és w_2 átlaga	2	0,5	0,5

(6)

A megfigyelt SN V SN alaknak potenciálisan két w felelhet meg. Ők alkotják a W halmazt. Az egyes megszorítások súlyozott átlaga értelmezhető ezen a W halmazon: valamely p_w súlyok mellett

$$C_i(W) = \sum_{w \in W} p_w \cdot C_i(w), \qquad \text{ahol} \qquad \sum_{w \in W} p_w = 1. \tag{7}$$

A (6) táblázatban a W halmaz mindkét elemére $p_w = 0,5$. Ha ezt az utolsó, átlagolt sort hasonlítjuk össze l sorával, arra a következtetésre jutunk, hogy *Neg mellett V-right is l-t preferálja, és mindkettőt lejjebb kell rangsorolni. Ráadásul V-left szempontjából pedig W a jobb, magasabban lenne a helye. Így tehát az algoritmus immár fel fogja tudni cserélni V-right és V-right rangsorolását. Vagyis a tanuló eljuthat a tanító nyelvtanához; de legalábbis egy azzal ekvivalens rangsoroláshoz, amelyben bár a megszorítások sorrendje eltérhet, de amely a célnyelvvel azonos nyelvet határoz meg.

A *szétválasztás menetrendje* a következőképpen módosul az ily módon bevezetett *Általánosított Robusztus Interpretatív Parszolás* nevű eljárásban [3]:

Minden $C_i \in \mathsf{Con}$ megszorításra, és valamely p_w értékek mellett,
1. ha $C_i(W) > C_i(l)$, akkor a C_i megszorítás l-t preferálja;
2. ha $C_i(W) < C_i(l)$, akkor a C_i megszorítás W-t preferálja.

Egyetlen kérdés maradt megválaszolatlanul: mi határozza meg a p_w értékeket a (7) képletben? Két közelmúltbeli cikkemben két különböző megoldást javasoltam. Egyiket a szimulált hőkezelés (szimulált lehűtés; *simulated annealing*), a másikat pedig a genetikai algoritmusok (*genetic algorithms*) ihlették.

4.1. GRIP: szimulált hőkezelés

A tanulás elején nem bízhatunk a tanuló nyelvtanában, mert az meglehetősen különbözhet a célnyelvtantól. Ha azonban hiszünk a tanulás sikerében, akkor fokozatosan növelhetjük a tanuló nyelvtanába vetett bizalmunkat. Ezért a tanulás elején a p_w súlyokat egyenlően szeretnénk elosztani W elemei között, a maximum-entrópia módszerek mintájára. A tanulás végén pedig oly módon, hogy csak a tanuló nyelvtana által legjobbnak tartott W-beli elem kapjon nullától különböző súlyt. Az utóbbi eset azonos a Tesar és Smolensky-féle eredeti RIP eljárással.

A *GRIP algoritmusnak* nevezett javaslatom [3] lényege az, hogy vezessünk be egy Boltzmann-eloszlást W-n. Ha $H(w)$ valós értékű, mint például a harmónia-nyelvtanban, akkor a Boltzmann-eloszlás alakja jól ismert:

$$p_w = \frac{e^{-H(w)/T}}{Z(T)}, \qquad \text{ahol} \qquad Z(T) = \sum_{w \in W} e^{-H(w)/T} \tag{8}$$

A termodinamikából kölcsönzött Boltzmann–Gibbs eloszlást egy pozitív T paraméter („hőmérséklet") jellemzi. Ha T nagyon magas ($T \gg H(w)$ minden $w \in W$-re), akkor a p_w súlyok (közel) egyenlően oszlanak el W elemei között. Ha viszont T nagyon alacsony ($0 < T \ll H(w)$), akkor a súly nagy része a legalacsonyabb $H(w)$ „energiájú" elem(ek)re koncentrálódik. Az optimálistól eltérő W-beli elemek p_w értékei nullához tartanak. A *szimulált hőkezelés* (szimulált lehűtés) név alatt ismert eljárások lényege az, hogy az algoritmus T paramétere nagyon magas értékről nagyon alacsony értékre fokozatosan csökken le.

A szimulált hőkezelés optimalizációs eljárásként ismert, és korábban ekként alkalmaztam az OT-ban is. Az *SA-OT algoritmus* egy performancia-modell: egy heurisztikus módszer az optimális jelölt megkeresésére [21,8,7]. Most azonban nem az optimális jelöltet keressük, hanem nyelvtant tanulunk.

Az *Általánosított Robusztus Interpretatív Parszolás* eljárás újítása az, hogy nem egyetlen w viselkedését veti össze az l viselkedésével megszorításonként, hanem az összes lehetséges W-beli jelölt viselkedésének súlyozott átlagát. A p_w súlyokat kell tehát meghatároznunk, és *erre* használjuk a Boltzmann-eloszlás (8) képletét. Arra tehát, hogy az egyes megszorítások W-n vett súlyozott átlagát definiáló (7) képletben szereplő p_w súlyokat kiszámítsuk. Majd, a tanulás során fokozatosan csökkentjük a (8)-ban használt T értékét, és ezáltal módosulnak a súlyok is. Kezdetben W minden eleme hozzájárul a megszorítások átrangsorolásának meghatározásához. Később azonban csak azok a jelöltek, amelyek a tanuló nyelvtana szerint a legharmonikusabbak W-ben.

Az algoritmusból azonban egy csavar még hiányzik. A (8) képlet valósértékű $H(w)$ függvényt feltételez. De az optimalitáselméletben $H(w)$ vektorértékű, amint azt (3) alatt láttuk. Ezért az idézett cikkemben a (8) Boltzmann-eloszlást vektorértékű $H(w)$-ra is értelmeznem kellett. Az eredmény formailag sok szempontból hasonlít az SA-OT algoritmusra. A Boltzmann-eloszlás T „hőmérséklet" paraméterének szerepét egy (K, t) paraméterpár veszi át, és ezek határozzák meg a p_w súlyokat. Az eljárás mögött húzódó matematikai gondolatmenet, valamint a pszeudokód és annak elemzése megtalálható [3]-ben – itt hely hiányában nem térhetünk ki ezekre a részletekre.

Ha a (K, t) paraméter már a tanulási folyamat elején is nagyon alacsony, akkor visszajutunk a hagyományos RIP eljáráshoz. Vajon a GRIP algoritmussal, magasabb (K, t) kezdőértékek mellett, javítható a tanulás sikeressége?

4.2. JRIP: „genetikai algoritmus"

[4] egy másik – matematikailag egyszerűbb – megközelítést mutat be a p_w súlyok meghatározására. Az alfejezet címében szereplő idézőjelek arra utalnak, hogy az alábbiakban leírtak csak távolról emlékeztetnek a genetikai algoritmusokra: nincs mutáció és szelekció, csupán egy változó összetételű rangsorolás-populáció, amely, remélhetőleg, konvergál a „megoldás" felé.

Yang [22] gondolatát követve, a javaslat lényege az, hogy a tanuló nem egy, hanem r darab nyelvtannal (esetünkben megszorítás-rangsorolással) rendelkezik. Ezeket külön-külön, véletlenszerűen inicializáljuk, és külön-külön tanulnak a RIP algoritmus szerint. A k-ik hierarchia ($1 \leq k \leq r$) minden egyes bejövő adat után kiszámítja a maga l_k és w_k jelöltjeit: ő maga mely jelöltet választaná, illetve a megfigyelt alak mely interpretációját találja optimálisnak. Ha ezek után a k-ik hierarchia összehasonlítja l_k-t w_k-val, lejjebb sorolja az l_k-t preferáló megszorításokat, és feljebb sorolja a w_k-t kedvelőket, akkor visszajutunk a hagyományos RIP algoritmushoz. Ha nem is mindegyik nyelvtan, de valamelyik közülük előbb-utóbb a célnyelvtanhoz fog konvergálni.

Ez a megközelítés azonban nem lenne plauzibilis gyermeknyelv-elsajátítási modell. Mind a k hierarchia csak kis valószínűséggel fog egyszerre sikerrel járni [4]. Ha pedig a nyelvtanok egy része nem jut el a célnyelvtanhoz, akkor a felnőttek honnan tudják, hogy melyik nyelvtant kell használniuk? A teljes nyelven tesztelik valamennyi nyelvtant? Számítógépes kísérletek játéknyelvtanai esetén egy ilyen teszt még elképzelhető lenne, de nem valódi nyelv esetén.

Ezért javasolom, hogy az egyes hiararchiák a saját maguk által optimálisnak tartott l_k jelöltet ne a saját maguk által meghatározott w_k jelölthöz hasonlítsák, hanem valamennyi w_k „átlagához". A rangsorolások a *hierarchiák populációjában* közösen interpretálják a bejövő alakot, hátha közös erővel sikeresebbek, mint egyenként. Közösen határozzák meg azt a $C_i(W)$ határértéket, amellyel utána mindenki külön-külön összeveti a saját $C_i(l_k)$-ját, hogy eldöntse, lejjebb vagy feljebb rangsorolja-e a C_i megszorítást a saját hierarchiájában. Sikeres tanulás esetén mind az r rangsor a célnyelvtanhoz konvergál.

Így jutunk el a *JRIP algoritmushoz*. A (7) képlet a következő alakot veszi fel:

$$C_i(W) = \frac{1}{r} \sum_{k=1}^{r} C_i(w_k) \tag{9}$$

Másképp megfogalmazva, a (7) egyenletbeli p_w arányos azon populációbeli nyelvtanok számával, amelyek w-t választották w_k gyanánt a W halmazból.

Az $r = 1$ eset megfelel a hagyományos RIP algoritmusnak. Vajon növelhető a tanulás sikere JRIP-pel, ha magasabb r-t választunk?

5. Szóhangsúly

A tagadó mondat eddig tárgyalt szórendjéhez hasonló problémával szembesül a tanuló (algoritmus) a hangsúly elsajátításánál is. A szóhangsúly kurrens fonológiai elméletei a szótagokat *lábakba* szervezik, de ezek nem „hallhatóak". Következésképp a tanuló nem tudhatja, hogy például a *hókusz-pòkusz* négyszótagú szó jambikus vagy trochaikus nyelvre bizonyíték-e. Elemezhető akár [*hók*][*uszpòk*]*usz*-ként, akár [*hókusz*][*pòkusz*]-ként. A szóhangsúly példáján mutatta be [2] a RIP algoritmust, és ezért én is ezen a példán illusztálom, hogy az általam javasolt két új módszer mennyit képes javítani a RIP algoritmuson.

A metrikus fonológia szerint a szótagok metrikus lábakba szerveződhetnek. Egy láb egy vagy két szótagból állhat. Az egyik láb kiemelt: a „feje" kapja a szó

főhangsúlyát. A többi láb feje mellékhangsúlyt kap. A két szótagból álló lábak másik szótagja, valamint a lábakon kívül eső szótagok nem kapnak hangsúlyt. A metrikus fonológia OT modelljeiben a megszorítások vonatkozhatnak a szótagokra (például nehéz szótag kapjon hangsúlyt; ne kerüljön szótag a lábakon kívülre), a lábakra (például a láb legyen kétszótagú; a láb legyen jambikus) és az egész szó szerkezetére (például a szó bal határa essen egybe egy láb bal határával). Kísérleteim során ugyanazt az OT metrikus fonológiai szakirodalomban széles körben elterjedt tizenkét megszorítást használtam, mint Tesar és Smolensky [2].

A kísérlet elején mind a tanító, mind a tanuló nyelvtanát véletlenszerűen inicializáltam. A tizenkét megszorításhoz egy-egy 0 és 50 közötti lebegőpontos rangsorértéket rendeltem, Boersma és Magri algoritmusainak megfelelően [16,18], eltérően az eredeti *EDCD* algoritmustól [14,2]. Minél magasabb egy megszorítás rangsorértéke, annál magasabbra kerül a hierarchiában. Négy algoritmust vizsgáltam: Boersma *GLA*-je az l-t preferáló megszorítások rangsorértékét 1-gyel csökkenti, és a W-t preferáló megszorításokét 1-gyel növeli. Magri algoritmusa a legmagasabbra rangsorolt, l-t preferáló megszorítás rangsorértékét 1-gyel csökkenti, és az összes – n darab – W-t preferáló megszorításét $1/n$-nel növeli. Az Alldem algoritmus csak az l-t preferáló megszorításokhoz nyúl, míg a Topdem algoritmus kizárólag a legmagasabbra rangsorolt, l-t preferáló megszorítás rangsorértékét csökkenti (szintén 1-gyel).

A nyelvtanuló feladata egy négy szóból álló lexikon helyes hangsúlyozásának a megtanulása volt. A lexikon szavai négy és öt, könnyű és nehéz szótagokból álltak: *ab.ra.ka.dab.ra*, *a.bra.ka.da.bra*, *ho.kusz.po.kusz* és *hok.kusz.pok.kusz*. A tanító ezeket látta el szóhangsúllyal a saját nyelvtana szerint, majd törölte a lábhatárokat, és az így generált nyelvi adatokat ismételgette a tanulónak. A tanulás akkor volt sikeres, ha a tanuló talált olyan hierarchiát, amellyel reprodukálta az általa megfigyelt nyelvi adatokat. Egy-egy paraméterbeállítás mellett a kísérletet több ezerszer megismételtem, és mértem a sikeres tanulások arányát.

Amikor a GRIP és a JRIP paraméterei a hagyományos RIP-nek feleltek meg, a sikeres tanulás aránya 76-78% körül volt, az algoritmus részleteitől függően. Megfelelő paraméterbeállításokkal azonban ez az arány jóval 90% fölé – néhány további trükkel pedig akár 95% fölé is – emelkedett [3,4]. A különbség statisztikailag erősen szignifikáns, bizonyítván a GRIP és JRIP algoritmusok sikerét.

6. Összefoglalás és utószó

Bemutattam, hogy az OT tanulóalgoritmusok milyen problémával szembesülnek, ha a tanítóadatok nem tartalmaznak minden fontos információt. A megfigyelhető adat lehetséges értelmezései közül a hagyományos RIP eljárás a tanuló nyelvtana szempontjából legjobbat választja. Ehelyett az értelmezések megszorításssértései átlagolását javasoltam, két különböző módszerrel. A szóhangsúllyal folytatott kísérletek során mindkét módszer szignifikánsan javított a RIP hatékonyságán.

A konferenciaabsztrakt megírása óta eltelt két hónap. Kislányom időközben elsajátította az /e/ és az /i/ közötti fonemikus különbséget a magyar nyelv nyelvtanában. Vajon milyen tanulóalgoritmust használt?

Hivatkozások

1. Prince, A., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell, Malden. Eredetileg: *Technical Report nr. 2. of the Rutgers University Center for Cognitive Science* (RuCCS-TR-2) (1993/2004)
2. Tesar, B., Smolensky, P.: Learnability in Optimality Theory. MIT Press, Cambridge, MA – London (2000)
3. Biró, T.: Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory. Journal of Logic, Language and Information (accepted)
4. Biró, T.: Uncovering information hand in hand: Joint Robust Interpretive Parsing in Optimality Theory. Linguistic Inquiry (submitted)
5. Smolensky, P., Legendre, G., eds.: The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. MIT Press (2006)
6. Rebrus, P.: Optimalitáselmélet. In Siptár, P., ed.: Szabálytalan fonológia. Tinta Könyvkiadó, Budapest (2001) 77–116
7. Bíró, T.: Finding the Right Words: Implementing Optimality Theory with Simulated Annealing. PhD thesis, University of Groningen (2006) ROA-896.
8. Bíró, T.: A sz.ot.ag: Optimalitáselmélet szimulált hőkezeléssel. In Alexin, Z., Csendes, D., eds.: III. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SzTE Informatikai Tanszékcsoport (2005) 29–40
9. Ellison, T.M.: Phonological derivation in Optimality Theory. In: Proceedings of the 15th CoLing Conference. Volume 2. (1994) 1007–1013
10. Eisner, J.: Efficient generation in primitive Optimality Theory. In: Proceedings of the 8th conference of EACL. (1997) 313–320
11. Tesar, B., Grimshaw, J., Prince, A.: Linguistic and cognitive explanation in Optimality Theory. In Lepore, E., Pylyshyn, Z., eds.: What is Cognitive Science? Blackwell, Malden, MA (1999) 295–326
12. Eisner, J.: Easy and hard constraint ranking in Optimality Theory: Algorithms and complexity. In Eisner, J., Karttunen, L., Thériault, A., eds.: Finite-State Phonology: Proc. of the 5th SIGPHON Workshop, Luxembourg (2000) 57–67
13. Tesar, B.: Computational Optimality Theory. PhD thesis, University of Colorado, Boulder (1995) ROA-90.
14. Tesar, B., Smolensky, P.: Learnability in Optimality Theory. Linguistic Inquiry **29**(2) (1998) 229–268
15. Boersma, P.: How we learn variation, optionality, and probability. Proceedings of the Institute of Phonetic Sciences, Amsterdam (IFA) **21** (1997) 43–58
16. Boersma, P., Hayes, B.: Empirical tests of the Gradual Learning Algorithm. Linguistic Inquiry **32** (2001) 45–86 ROA-348.
17. Boersma, P.: Some correct error-driven versions of the Constraint Demotion algorithm. Linguistic Inquiry **40**(4) (2009) 667–686
18. Magri, G.: Convergence of error-driven ranking algorithms. Phonology **29**(2) (2012) 213–269
19. Lopopolo, A., Biró, T.: Language evolution and SA-OT: The case of sentential negation. Computational Linguistics in the Netherlands J **1** (2011) 21–40
20. Samek-Lodovici, V., Prince, A.: Optima. ROA-363 (1999)
21. Bíró, T.: How to define Simulated Annealing for Optimality Theory? In: Proceedings of Formal Grammar/Mathematics of Language, Edinburgh (2005)
22. Yang, C.D.: Knowledge and Learning in Natural Language. Oxford University Press, Oxford, UK (2002)

II. Lexikológia, fordítás

Angol nyelvű összetett főnevek értelmezése parafrázisok segítségével

Dobó András[1], Stephen G. Pulman[2]

[1] Szegedi Tudományegyetem, Informatikai Tanszékcsoport,
H-6720 Szeged, Árpád tér 2.
dobo@inf.u-szeged.hu
[2] University of Oxford, Department of Computer Science,
Wolfson Building, Parks Road, Oxford, OX1 3QD, Egyesült Királyság
stephen.pulman@cs.ox.ac.uk

Kivonat: Az angol nyelvben gyakran használnak összetett főneveket, melyek jelentésének meghatározása számos számítógépes nyelvészeti probléma megoldásának fontos eleme. Egy olyan módszert mutatunk be cikkünkben, mely alkalmas két szóból álló angol nyelvű összetett főnevek értelmezésére parafrázisok segítségével, ahol parafrázisok alatt igéket és elöljárószavakat értünk. Ez a módszer először megfelelő parafrázisokat keres statikus korpuszokban, majd webes kereséseket alkalmaz a helytelen parafrázisok kiszűrésére. A módszerünk által visszaadott parafrázisokat angol anyanyelvű személyekkel értékeltettük ki. Az első, második, illetve harmadik helyen visszaadott parafrázisokra rendre átlagosan 3,1842, 2,7687, illetve 2,5583 pontot adtak az értékelők megfelelősségük alapján (1-től 5-ig terjedő skálán), ami véleményünk szerint biztató eredmény a feladat nehézségét figyelembe véve.

1 Bevezetés

Mind az írott, mind a beszélt angolban bőségesen előfordulnak összetett főnevek (noun compound), melyek Downing [1] definíciója alapján főnevek olyan sorozatai, melyek egy főnévként viselkednek (az angol nyelvben az összetett főneveket külön kell írni). Értelmezésük, különösen gyakori használatuk miatt, nélkülözhetetlen számos számítógépes nyelvészeti probléma megoldásához, mint például a gépi fordításhoz és információ-visszakereséshez. Például amikor egy információ-visszakereső rendszer a *plastic bottles* (műanyag palackok) kifejezéshez keres információkat, akkor szükséges tudnia, hogy a *bottles that are made of plastic* (műanyagból készült palackok) kifejezésről talált információ releváns-e.

Első gondolatra statikus szótárak használata megfelelőnek tűnik e feladat megoldására, azonban még a gyakran használt összetett főnevekre is kis lefedettséget adnak e szótárak [2], és az összetett főnevek gyakorisági spektruma Zipf-eloszlást mutat [3], vagyis a legtöbb összetett főnévnek nagyon ritka az előfordulása.

E kutatás célja a két szóból álló angol nyelvű összetett főnevek automatikus értelmezése statikus korpuszok segítségével. Wright [4] és Nakov és Hearst [5] nyomán úgy gondoljuk, hogy az összetett főnevek parafrázisokkal (paraphrase - igék és elöljá-

rószavak) történő értelmezése célravezetőbb, mint korlátozott számú absztrakt kategória alkalmazása, mivel lényegében végtelen különböző összetett főnév létezik és finom jelentésbeli különbségek kifejezésére is képesek. Továbbá úgy gondoljuk, hogy parafrázisok egy sorrendbe állított listája alkalmasabb e szószerkezetek értelmezésére mint egyetlen parafrázis, mivel egy gyakran nem elég egy összetett főnév teljes jelentéskörének megadására. Például, a *malaria mosquito* (malária moszkitó) egy lehetséges értelmezése a következő sorrendbe állított parafrázis lista lehetne:

1. carry (hordoz)
2. spread (terjeszt)
3. be infected with (által fertőzött)

, mivel *a malaria mosquito is a mosquito that carries / spreads / is infected with malaria* (a malária moszkitó egy olyan moszkitó, ami maláriát hordoz / maláriát terjeszt / malária által fertőzött).

A kidolgozott módszer olyan parafrázisokat keres a felhasznált statikus korpuszban, melyek alkalmasak az input összetett főnév értelmezésére. A módszer alapja az, hogy megkeresi azokat a mondatokat a korpuszban, amelyek egy parafrázis segítségével mondatba foglalják az adott összetett főnevet, megszámlálja, hogy az egyes parafrázisok hányszor fordultak elő a szókapcsolattal, majd e gyakoriságok alapján létrehoz egy rendezett listát. Ezt az alapötletet később több módon kibővítettük. Algoritmusunkat korábban angol nyelven már bemutattuk a Dobó és Pulman [6] cikkben.

2 Kapcsolódó munkák

2.1 Kategóriaalapú módszerek

Vannak olyan nyelvészeti elméletek, mint például Levié [7], melyek szerint az összetett főnevek mindegyike besorolható kis számú kategóriák valamelyikébe a főnevek között fennálló szemantikai kapcsolat alapján. Sok korábbi összetett főnév értelmezési módszer ezeken az elméleteken alapszik, és ennek megfelelően az összetett főneveket absztrakt kategóriákba sorolással próbálja meg értelmezni.

Rosario és Hearst [8] például 18 absztrakt osztály használatát indítványozza és egy olyan általános gépi tanulási módszert alkalmaz biomedikai összetett szavak osztályozására, mely doménspecifikus lexikai hierarchiával rendelkezik.

Nastase és Szpakowicz [9] szintén gépi tanulási módszereket alkalmazó algoritmust publikált összetett szavak klaszterezésére. Ehhez a WordNetből és a Roget's Thesaurusból kinyert tulajdonságokat használtak, és 30 klasztert definiáltak, melyek 5 szuperklaszterbe tartoztak.

Azonban az ebbe a csoportba tartozó módszereket számos kritika érte. Habár megvan az az előnyük, hogy megragadják az összetett főnevekben megtalálható általános kapcsolatokat, az általuk felhasznált kis számú kategória korlátozza is őket [2]. Downing [1] az egyike azoknak, akik leginkább kritizálják ezeket a módszereket. Szerinte olyan sokféle összetett főnévi kapcsolat létezik, hogy azt felsorolni lehetetlen, és nagyon sok olyan kapcsolat van ezek között, mely egyetlen általánosan használt kapcsolati kategóriába sem illeszkedik bele. Véleménye szerint az is problémát okoz, hogy mivel a használt kategóriák száma limitált, ezért a kategóriák homályosak, többértel-

műek lehetnek, és így különböző belső kapcsolattal rendelkező összetett főnevek is azonos kategóriákba kerülhetnek. Továbbá azt is nehéz lenne megállapítani, hogy a kategóriáknak mely halmaza lenne a legmegfelelőbb az összetett szavakban megtalálható kapcsolatok osztályozására, mivel a kimondottan összetett szavakkal foglalkozó nyelvészek sem értenek egyet a még fő kategóriákban sem [10].

2.2 Parafrázisalapú módszerek

Az előző alfejezetben említett problémák egy lehetséges megoldása az, ha parafrázisokat, vagyis igéket és elöljárószavakat, használunk az összetett szavak értelmezésére előre definiált absztrakt kategóriák helyett. Parafrázisok használata esetén a lehetséges kapcsolati kategóriák számát csak az adott nyelv szókincse korlátozza, továbbá még nagyon finom jelentésbeli különbségeket is ki lehet velük fejezni, valamint nincs egyetlen olyan összetett főnév sem, amely egyetlen kategóriába sem illik bele [2]. Ezért a parafrázis alapú módszerek az elmúlt években egyre népszerűbbek lettek.

Az egyik korai parafrázis alapú összetett szavakat értelmező módszert Laurer [10] fejlesztette ki. Ugyan parafrázisokkal dolgozik, mégis csak nyolc elöljárószót alkalmaz parafrázisként, ezért ez a módszer még inkább a kategóriaalapú módszerek családjába tartozik, és rendelkezik azok hátrányaival.

Ezzel szemben Nakov és Hearst [5], valamint Nakov [11] módszere már ténylegesen parafrázisalapú, az összetett szavak értelmezéséhez webes keresések által visszaadott szövegtöredékekből nyeri ki a parafrázisok listáját azok gyakoriságával együtt.

A SemEval-2 Workshop 9. feladatának [2] megoldására is született számos módszer. A feladatban adott összetett szavak egy listája és minden összetett szóhoz adott lehetséges parafrázisok egy halmaza. A cél olyan algoritmus írása volt, mely minden összetett szóhoz visszaadja a parafrázisok rendezett sorozatát, ahol a rendezés alapja az, hogy a parafrázisok mennyire megfelelőek az összetett szóhoz.

Erre a feladatra Nulty és Costello [12] egy olyan módszert dolgoztak ki, mely a tanító halmazból kinyert parafrázis gyakoriságokat használja fel úgy, hogy az általánosan használt parafrázisokat előnyben részesíti a kevésbé általánosakkal szemben.

A feladat megoldásához Wubbennek [13] teljesen más volt a stratégiája: egy osztályozó algoritmust hozott létre a WordNetből, a tanító halmazból és a Web 1T 5-gram Corpusból kinyert tulajdonságok alapján.

3 Módszerünk bemutatása

Célunk egy olyan módszer létrehozása volt, mely alkalmas tetszőleges két szóból álló angol nyelvű összetett főnév értelmezésére úgy, hogy ha bemenetként megkapja összetett főnevek egy listáját, akkor mindegyikhez visszatérjen parafrázisok egy rendezett listájával, igéket és elöljárószavakat használva parafrázisként.

Majdnem minden összetett szóban a második szó a fej (alaptag), míg az első az alárendelt tag, ami a fej egy tulajdonságát határozza meg. A két szó által alkotott összetett szó szintaktikailag úgy viselkedik, mint ahogy a feje [5], [10]. Munkánk során feltettük, hogy ez a tulajdonság az értelmezendő összetett szavakra fennáll, ezért módszereinkkel csak olyan parafrázisokat kerestünk, melyeknek alanya az összetett szó második főneve és tárgya az összetett szó első főneve.

3.1 A két alapmódszer

Az összetett szavakhoz megfelelő parafrázisok keresésére és kinyerésére két alapmódszert dolgoztunk ki.

Az alany-parafrázis-tárgy hármasokat alkalmazó módszer. Alapötletünk az volt, hogy oly módon tudunk megfelelő parafrázisokat találni egy összetett szóhoz, hogy ha egy statikus korpuszban keresünk olyan mondatokat, melyek egy parafrázis segítségével mondatba foglalják az adott összetett szót. Ehhez az algoritmus végigolvassa az alkalmazott korpuszt és megkeresi az összes olyan előforduló *(a, p, t)* hármast, melyben:

- *p* egy ige, melynek *a* az alanya és *t* a közvetlen tárgya
- *p* egy elöljárószavas ige, melynek *a* az alanya, az elöljáró az igével szorosan egybe tartozik (particle) és *t* az elöljárószavas ige közvetlen tárgya
- *p* egy elöljáró, ami *a*-nak egy módosítószava, és *t* a közvetlen tárgya az elöljárószónak

Ez a kinyerési módszer nagyon hasonló Nakov [11] módszerének ahhoz a részéhez, mely során a webes kereső által visszaadott, nyelvtanilag elemzett szövegtöredékekből kinyeri a tulajdonságokat az összetett szavakhoz.

Ez után a parafráziskinyerési módszer után módszerünk minden egyes bemeneti összetett főnévhez megkeresi azokat az *(a, p, t)* hármasokat, ahol *t* az összetett szó első, *a* pedig a második főneve. Ennek eredményeképpen megkapjuk parafrázisok egy listáját minden összetett főnévhez, az összetett főnév és a parafrázis együttes előfordulási gyakoriságával együtt. Ez az együttes előfordulási gyakoriság lesz a parafrázis pontszáma az adott összetett szóhoz. Például, ha 50 darab *(a=story, p=be about, t=adventure)* hármast talál az algoritmus, akkor az *adventure story* összetett főnév *be about* parafrázisához 50-es pontszámot rendel.

Ugyan az e módszerünk által megtalált parafrázisok általában megfelelőek voltak, nagyon kevés parafrázist talált az algoritmus még gyakori összetett főnevek esetén is, mivel az összetett szavak ritkán voltak ilyen módon mondatba foglalva. Így kipróbáltunk egy másik módszert is, mely a precision rovására magasabb recallal rendelkezik.

Az alany-parafrázis és parafrázis-tárgy párokat használó módszer. Ennek a módszernek az alapötlete az, hogy ha létezik olyan parafrázis, melynek a vizsgált összetett szó második főneve gyakran az alanya és első főneve gyakran a tárgya, akkor nagy esély van arra, hogy ez a parafrázis alkalmas az összetett szó értelmezésére. Ezért ez a módszer a korpusz végigolvasása közben azokat az *(a, p)* párokat keresi meg, melyekben:

- *p* egy ige, melynek *a* az alanya
- *p* egy elöljárószavas ige, melynek *a* az alanya és az elöljáró az igével szorosan egybe tartozik (particle)
- *p* egy elöljáró, ami *a*-nak egy módosítószava

Továbbá megkeresi azokat a *(p, t)* előfordulásokat is, melyekben:

- *p* egy ige, melynek *t* a közvetlen tárgya
- *p* egy elöljárószavas ige, melyben az elöljárószó az igével szorosan egybe tartozik (particle) és *t* az elöljárószavas ige közvetlen tárgya
- *p* egy elöljárószó, aminek *t* a közvetlen tárgya

E párok kinyerése után az algoritmus olyan *(a, p)* és *(p, t)* párokat keres egy összetett főnévhez, melynek második szava *a* és első szava *t*. Ez két parafrázislistát eredményez, egyet a második főnévhez (alanyhoz), egyet pedig az első főnévhez (tárgyhoz). Ebből a két listából egy olyan *(a, p, t)* listát kell létrehoznia, mely rangsorolja a parafrázisokat az összetett szó értelmezésére való alkalmasságuk szerint. Ehhez megkeresi azokat a parafrázisokat, melyek mindkét listában szerepelnek, és ezeket belerakja a közös listába, egy, a két listában talált gyakoriságból számolt pontszámmal.

Azonban szimplán gyakoriságok használata itt nagyon nagy problémát jelent: attól függetlenül, hogy az összetett szó első (tárgy) vagy második (alany) főnevét tekintjük, a hozzá megtalált leggyakoribb parafrázisok olyan nagyon gyakori igék, mint a *be*, a *do* vagy a *make*. Ezért a kombinált listában is ezek az igék szerepelnének legmagasabb pontszámmal, és ezek egyike sem jellemzi jól az összetett szavakat. Azért, hogy ezt elkerüljük, mind az *(a, p)* és *(p, t)* párok esetén pontonkénti kölcsönös információt [14] használtunk a gyakoriságok helyett. Az *(a, p)* és *(p, t)* párok pontonkénti kölcsönös információját ezután az algoritmus összeszorozza, és a parafrázisok ezzel a pontszámmal kerülnek be a közös *(a, p, t)* listába.

Például, ha az *(a=bottle, p=be for)* párnak és a *(p=be for, t=water)* párnak rendre 40 és 50 a gyakorisága, a *bottle* szó 500-szor és a *be for* kifejezés 2000-szer fordul elő *(a, p)* párban, valamint a *water* szó 800-szor és a *be for* kifejezés 1500-szor fordul elő *(p, t)* párban, továbbá az algoritmus összesen 2000000 *(a, p)* párt illetve 1500000 *(p, t)* párt talál, akkor a *be for* parafrázis *water bottle* szóhoz vett pontszáma 37,7153 lesz ezzel a módszerrel.

Mivel a 0 értéknél kisebb pontonkénti kölcsönös információ negatív asszociációt (disszociációt) jelent, ezért csak azokat a parafrázisokat vettük figyelembe, melyek esetén az *(a, p)* és a *(p, t)* pár is pozitív pontonkénti kölcsönös információval rendelkezik. Továbbá, mivel a pontonkénti kölcsönös információ instabil kis gyakoriságok esetén [14], ezért az 5-nél kisebb *(a, p)* vagy *(p, t)* gyakorisággal rendelkező parafrázisokat nem vettük figyelembe.

Azért, hogy módszereink hatékonyabban működjenek, mindkét módszer esetén az összes szót lemmatizáltuk, és a keresést is az összetett főnevek szavainak lemmájával végeztük. A szavak lemmáját a WordNet segítségével határoztuk meg.

3.2 A felhasznált korpuszok és azok előfeldolgozása

A parafrázisok kereséséhez a British National Corpust és a Web 1T 5-gram Corpust használtuk fel. Azért, hogy a megfelelő *(a, p)* és *(p, t)* párokat, illetve *(a, p, t)* hármasokat az algoritmusok ki tudják nyerni, szükséges a korpusz szavai között fennálló nyelvtani kapcsolatok azonosítása. Ehhez a British National Corpusnak egy a C&C CCG automatikus nyelvtani elemzővel [15] feldolgozott példányát használtuk fel, melyben így a nyelvtani kapcsolatok már explicit módon adottak voltak.

A rendelkezésünkre álló Web 1T 5-gram Corpus azonban nem volt még nyelvtanilag elemezve. Az automatikus nyelvtani elemzéshez szükséges idő hiányában egy alternatív megoldást választottunk. A korpuszt szófajilag elemeztük a C&C CCG automatikus szófaji elemzővel, majd szófaji minták alapján próbáltunk a szavak között fennálló nyelvtani kapcsolatokra következtetni. Például, ha egy 4-gram a *főnév ige névelő főnév* szófaji mintával rendelkezik, akkor nagy annak az esélye, hogy az első *főnév* az *ige* alanya, míg a második *főnéve* az *ige* tárgya. Ezt és ehhez hasonló mintákat használtunk fel a nyelvtani kapcsolatok kinyerésére a Web 1T 5-gram Corpus esetén. Mivel a rövid szövegtöredékek automatikus szófaji elemzése nagy hibával jár, ezért csak a 4- és 5-gramokat használtuk fel.

3.3 Elöljárószavak

Az elöljárószóval rendelkező parafrázisokat különlegesen kezeltük az alany-parafrázis és parafrázis-tárgy párokat használó modell esetében: ha a modellünk egy ilyen parafrázist talál, akkor két *(a, p)* párt nyer ki a szövegből. Egy olyat, amelyben a parafrázis tartalmazza az elöljárószót, és egy olyat is, amelyben nem. Az elöljárószó nélkülit azért, mert egy olyan mondatból, mint a *"The professor teaches at a university"* logikusnak látszik az *(a=professor, p=teach)* pár kinyerése. Így ha például van egy *(p=teach, t=anatomy)* párunk is, akkor a két párt összekapcsolva megkaphatjuk a *teach* parafrázist az *anatomy professor* összetett szóhoz. Az is szükséges, hogy módszerünk kinyerjen egy *(a, p)* párt az elöljárószóval együtt is, mivel egyébként nem lenne képes elöljárószót tartalmazó parafrázisok megtalálására egyetlen összetett főnév esetében sem. A *(p, t)* párok és *(a, p, t)* hármasok esetén nincs szükség speciális bánásmódra.

3.4 Passzív parafrázisok

A passzív parafrázisok abban különböznek a többi parafrázistól, hogy látszólagos alanyuk valójában a cselekvés tárgya. Ezért egy olyan (a, p_1) párnak, melyben p_1 egy elöljárószó nélküli passzív parafrázis, lényegében ugyanaz a jelentése (legalábbis a mi szempontunkból), mint egy olyan (p_2, t) párnak, melyben $a=t$ és p_2 a p_1 parafrázis aktív alakja. Ezért logikus lenne az ilyen, lényegében azonos jelentésű párokat együtt kezelni, gyakoriságukat közösen számolni. Ennek érdekében ha algoritmusunk egy olyan (a, p_1) párt talál, melyben p_1 parafrázis elöljárószó nélküli és passzív, akkor ezt egy olyan (p_2, t) párként menti el, melyben $a=t$ és p_2 a p_1 parafrázis aktív alakja. Például a *"The pizza was eaten"* mondatból az alany-parafrázis és parafrázis-tárgy párokat használó modellünk a *(p=eat, t=pizza)* párt nyeri ki. Mivel a passzív parafrázisoknak nem lehetnek közvetlen tárgyai, ezért nem létezhetnek olyan *(p, t)* párok és *(a, p, t)* hármasok, melyekben *p* egy elöljárószó nélküli passzív parafrázis.

Azoknál a passzív parafrázisoknál pedig, melyek tartalmaznak egy olyan *by* elöljárószót, melynek van közvetlen tárgya, ez a tárgy valójában a cselekvés alanya. Ezért egy olyan (a_1, p_1, t_1) hármas, melyben a p_1 parafrázis passzív és tartalmazza a *by* elöljárószót, lényegében ugyanolyan jelentéssel bír, mint egy olyan (a_2, p_2, t_2) hármas, ahol $a_2=t_1$, $t_2=a_1$ és p_2 a p_1 parafrázis aktív alakja elöljárószó nélkül. Tehát az ilyen, lényegében azonos jelentésű hármasokat is érdemes együtt kezelni, gyakoriságukat közösen számolni. Így például a *"The house was built by an architect"* mondatból az

alany-parafrázis-tárgy hármasokat használó módszerünk az *(a=architect, p=build, t=house)* hármast nyeri ki. Az olyan *(a, p)* és *(p, t)* párokat, melyekben *p* szintén egy passzív parafrázis a *by* elöljárószóval, az alany-parafrázis és parafrázis-tárgy párokat alkalmazó modellünk ehhez nagyon hasonlóan kezeli. Az olyan passzív parafrázisokat, melyek a *by*-tól eltérő elöljárószót tartalmaznak, nem kell speciálisan kezelni.

A fent leírt átalakítások miatt azoknak az *(a, p, t)* hármasoknak, valamint *(a, p)* és *(p, t)* pároknak a gyakorisága, melyekben *p* egy passzív parafrázis a *by* elöljárószóval, az átalakított verzióikhoz lettek elmentve. Ezért, annak érdekében, hogy algoritmusunk ehhez hasonló parafrázisokat is megtalálhasson összetett főneveinkhez, mindkét alapmódszerünk keres aktív, elöljárószó nélküli parafrázisokat a megfordított összetett szóhoz is (melyben a főnevek sorrendje fel lett cserélve; lehet, hogy így nem egy tényleges főnevet kapunk, de ez számunkra most lényegtelen). Ha talál ilyen parafrázist, akkor annak a passzív, *by* elöljárószóval kiegészített változatát használja fel, a megtalált parafrázis gyakoriságával.

Vagyis, ha például a *band concert* összetett szóhoz keres az algoritmus passzív, *by* elöljárószót tartalmazó parafrázist, akkor az alany-parafrázis-tárgy hármasokat használó módszerünk a szövegből kinyert *(a=band, p, t=concert)* alakú hármasokat keres. Például az *a=band, p=give, t=concert)* hármas esetén az algoritmus elmenti a *be given by* parafrázist a *band concert* összetett szóhoz, a talált hármas pontszámát felhasználva. Ez a másik alapmódszerünk esetén is nagyon hasonlóan működik.

3.5 Ambitranzitív igék

Angolban az igék lehetnek szigorúan tárgyasak, szigorúan tárgyatlanok, illetve ambitranzitívak [16], ahol az utolsó kategóriába tartozó igék tárgyas és tárgyatlan igeként is funkcionálhatnak. Jó példa szigorúan tárgyas igére a *like* és a *recognise*, szigorúan tárgyatlanra az *arrive* és a *run*, és ambitranzitívre a *break* és a *read*. Perlmutter [17] Unaccusative Hypothesise szerint a tárgyatlan igék két csoportra bonthatók: az unakkuzatív igék azok, melyek látszólagos alanya valójában a cselekvés tárgya (például *arrive*), és az unergatív igék azok, melyek látszólagos alanya ténylegesen a cselekvés alanya (például *run*). Ehhez nagyon hasonlóan az ambitranzitív igéket is két csoportra oszthatjuk: a páciens alanyú ambitranzitív igék azok, melyek unakkuzatív módon viselkednek intranzitív esetben és az ágens alanyú ambitranzitív igék azok, melyek unergatív tulajdonságúak intranzitív esetben [18]. Egy tipikus páciens alanyú ambitranzitív ige a *break*: a *"the window broke"* kifejezés valójában azt jelenti, hogy *"someone or something broke the window"*. Egy gyakori ágens alanyú ambitransitive ige pedig a *read*, mivel a *"she reads"* kifejezésben *she* ténylegesen a cselekvés alanya.

Tehát páciens alanyú ambitranzitív igék intranzitív használatakor módszerünk a cselekvés tényleges tárgyát (ami a látszólagos alany) helytelenül a cselekvés alanyaként nyerné ki. Ez hibákat eredményezne az összetett szavak értelmezésében. Azonban megfigyelhetjük, hogy az intranzitív esetben használt páciens alanyú ambitranzitív igék pontosan úgy viselkednek, mint a passzív igék: látszólagos alanyuk valójában a cselekvés tárgya. Ezért ezeket az igéket ugyanolyan módon kezeljük algoritmusunkban, mint a passzív igéket, és ezzel a fent leírt problémát kiküszöböljük. A páciens alanyú ambitranzitív igék felismeréséhez a Levin [19] által megadott átfogó listát használtuk fel.

3.6 Szinonimák, hipernimák, testvér szavak és szemantikailag hasonló szavak használata a magasabb recall elérése érdekében

Ugyan az általunk felhasznált korpuszok viszonylag nagyok, alapalgoritmusaink még így sem találnak bennük sok összetett főnévhez parafrázist. Kim és Baldwin [20] hipotézisét követve mi is úgy véljük, hogy hasonló jelentéssel bírnak azon összetett főnevek, melyek egymáshoz szemantikailag hasonló szavakból állnak. Így annak érdekében, hogy az összetett szavak értelmezésénél magasabb recallt tudjuk elérni, nemcsak az eredeti összetett szavakhoz kerestünk parafrázisokat, hanem azok olyan módosított változataihoz is, melyekben valamelyik (esetleg mindkettő) szót helyettesítettük az eredeti szó egy szinonimájával, hipernimájával, testvér szavával vagy pedig egy hozzá szemantikailag hasonló szóval. A szavak szinonimáit, hipernimáit és testvér szavait a WordNetből nyertük ki, míg a szavakhoz szemantikailag hasonló szavakat Lin [21] pusztán statikus korpuszokat felhasználó módszerével határoztuk meg.

3.7 A helytelen parafrázisok kiszűrése webes keresések segítségével

Az összetett szavak értelmezésére a korpuszból kigyűjtött parafrázisok sajnos sokszor nem helyesek, különösen az alany-parafrázis és parafrázis-tárgy párokat használó módszerünk esetén, illetve akkor, ha az összetett szó szavait a módszer helyettesítheti a szavak szinonimáival, hipernimáival, testvér szavaival vagy a szóhoz szemantikailag hasonló szavakkal. Ezért algoritmusunkat kibővítettük egy második lépéssel is, mely segít annak eldöntésében, hogy a megtalált parafrázisok közül melyek helyes értelmezései az összetett főneveknek, így növelve az algoritmus által elért precisiont.

Ehhez a lépéshez úgy döntöttünk, hogy webes kereséseket alkalmazunk a Google és a Yahoo! keresőrendszerek segítségével. Feltettük, hogy ha egy parafrázis alkalmas egy adott összetett szó értelmezésére, akkor léteznie kell legalább néhány olyan weblapnak, mely mondatba foglalja az összetett szót a parafrázis segítségével. Ezért minden (összetett szó, parafrázis) párhoz webes kereséseket indítottunk, és a parafrázisokat a keresésekre visszaadott lapok számának segítségével újrarendeztük.

Először egyszerű kereséseket próbáltunk ki, hasonlókat a Nakov és Hearst [5] és Nakov [11] által használtakhoz: egy n_1 n_2 összetett szó és p parafrázis esetén az összes lehetséges *"n2Infl THAT p n1Infl"* alakú lekérdezéssel kerestünk a keresőrendszerben, ahol *n1Infl* és *n2Infl* rendre az n_1 és n_2 főnevek lehetséges ragozott, illetve ragozatlan alakjai lehetnek, a *THAT* pedig vagy egy üres szó vagy az egyike a következő három vonatkozó névmásnak: *that*, *which* és *who*. Egy adott (összetett szó, parafrázis) párhoz tartozó összes ilyen alakú lekérdezésre visszaadott lapok számát összegezve definiáltuk az (összetett szó, parafrázis) pár webes pontszámát.

Azonban még ezek a keresések sem adtak vissza minden helyes (összetett szó, parafrázis) párhoz találatot. Ezért ezeket a kereséseket kibővítettük. Egyrészt úgy, hogy az igei parafrázisok esetén nemcsak a jelen idejű alakjukat használtuk fel, hanem egyéb igeidejű alakjaival is keresést indítottunk. Továbbá olyan kereséseket is használtunk, melyek joker karaktereket (*), 0 és 9 közötti számút, is tartalmaztak. Ezeket a joker karaktereket a parafrázis (*p*) és az első főnév (*n1Infl*) közé raktuk.

Miután egy adott (összetett szó, parafrázis) párhoz elvégeztük a fent leírt webes kereséseket és azok segítségével meghatároztuk a pár webes pontszámát, a pár végle-

ges pontszámát az eredeti pontszámának és a webes pontszámának segítségével számoltuk ki a következőképpen:

$$pontszám_{végső} = \ln(pontszám_{eredeti} + 1) * \ln(pontszám_{web} + 1) \quad (1)$$

ahol $pontszám_{eredeti}$ a pár eredeti és $pontszám_{web}$ a pár webes pontszáma. Az algoritmus ezután a parafrázisokat végső pontszámuk segítségével rendezi sorba.

4 Eredmények

A módszerek kiértékeléséhez a SemEval-2 Workshop 9. feladatának tesztadathalmazát használtuk fel. Ennek a feladatnak a célja olyan algoritmusok írása volt, melyek képesek az összetett főnevekhez már előre megadott lehetséges parafrázisokat megfelelősségük szerinti sorrendbe rakni. A mi algoritmusunk e feladat megoldásánál többre képes, ugyanis nincs szüksége bemenetként a lehetséges parafrázisok egy listájára, hanem a lehetséges parafrázisokat automatikusan nyeri ki a felhasznált korpuszból. Mivel módszerünk nem használja fel bemenetként az összetett főnevekhez adott lehetséges parafrázisok listáját, így olyan parafrázisokat is visszaad, melyek nincsenek ezen a listán. Ez okból kifolyólag a feladathoz biztosított kiértékelőt nem tudtuk módszereink teljesítményének mérésére felhasználni.

Helyette megkértünk 5 angol anyanyelvű személyt, hogy segítsenek módszerünk kiértékelésében. Mindegyiküknek odaadtuk a módszerünk által a bemeneti összetett szavakra visszaadott (összetett szó, parafrázis) párosok listáját, és ők minden párhoz egy 1 és 5 közé eső pontszámot rendeltek, ami a parafrázis minőségét adta meg (1: egyáltalán nem megfelelő, 5: teljesen megfelelő).

A limitált emberi erőforrás miatt nem tudtuk módszerünk összes változatát a felkért személyekkel kiértékeltetni, ezért a módszereink különböző változatait először mi magunk értékeltük ki, és csak az általunk legjobbnak vélt eredményeket adtuk oda a felkért személyeknek. Továbbá, szintén a kiértékelést gyorsítandó okból csak a tesztadatbázis első 50 összetett szavát használtuk fel. Mivel úgy véljük, hogy néhány parafrázis teljesen elegendő egy összetett szó teljes jelentéskörének a leírásához, ezért minden összetett szóhoz a módszerünk által visszaadott parafrázisok közül a három legmagasabb pontszámmal rendelkezőt vettük figyelembe.

Saját teszteléseink során arra az eredményre jutottunk, hogy a legjobban egy kombinált módszer teljesített. Ez két módszer kombinációjával jött létre: az egyik nem használ helyettesítő szavakat a parafrázisok kereséséhez, míg a másik felhasználja a WordNetből kinyert testvér szavakat az összetett szó eredeti szavainak helyettesítésére. A kombinált módszer a két módszer által visszaadott parafrázisok listáját egyesíteti, miután a testvér szavakat is alkalmazó módszer által visszaadott parafrázisokat újrapontozza a következőképpen:

$$pontszám_{új} = \frac{pontszám_{eredeti} * pontszám_{legalacsonyabb,nincsHelyettesítés}}{pontszám_{legmagasabb,helyettesítésTestvérSzavakkal}} \quad (2)$$

ahol $pontszám_{eredeti}$ az (összetett szó, parafrázis) pár eredeti pontszáma, $pontszám_{legalacsonyabb,nincsHelyettesítés}$ a helyettesítő szavakat nem használó módszer által visszaadott parafrázisok közül legkisebb pontszámmal rendelkezőnek a

pontszáma és $pontszám_{legmagasabb,helyettesítésTestvérSzavakkal}$ a helyettesítésként testvér szavakat alkalmazó módszer által visszaadott parafrázisok közül a legmagasabb pontszámmal rendelkezőnek a pontszáma. Ez által az újrapontozás által a második módszer által visszaadott legjobb parafrázis pontszáma meg fog egyezni az első módszer által visszaadott legrosszabb parafrázis pontszámával. Az ugyanazon módszer által visszaadott parafrázisok pontszáma közti arány így nem változik meg, viszont a kombinálás e módja előtérbe helyezi az első, lényegesen magasabb precisionnel rendelkező módszer által visszaadott parafrázisokat. Ahol pedig az első módszer nem ad vissza a kiértékeléshez elegendő (legalább 3) parafrázist, ott a lista kiegészül a második módszer által visszaadott parafrázisokkal. A kombinált módszerek közül mindkettő alany-parafrázis-tárgy hármasokat alkalmazott és a Web 1T 5-gram Corpust használta fel parafrázisok keresésére.

Az egyesített lista létrehozása után a listában szereplő parafrázisok mindegyikét újrapontozza webes keresések segítségével, a 3.7. alfejezetben leírt módon. A különböző webes pontozási módszereket a SemEval-2 Workshop 9. feladatának teszthalmazán automatikusan kiértékeltük a feladathoz adott kiértékelő segítségével. Ez alapján az a webes keresési módszer érte el a legjobb eredményt, amelyik a Google keresőrendszert, az igéknek csak a jelen idejű alakját és 0 és 1 közötti darabszámú joker karaktert használ, továbbá a keresésekben nem alkalmaz vonatkozó névmásokat.

Mielőtt a felkért személyek által visszaadott értékelésekből következtetéseket vontunk le, szükséges volt a személyek értékelésben való egyetértésének az igazolása. Amennyiben az értékelő személyek közt jelentős az egyet nem értés, akkor az általuk adott értékelés nem megbízható, és abból következtetéseket nem lehet levonni. Az adatok megbízhatóságának vizsgálatára Krippendorff [22] alfa metrikáját alkalmaztuk. A megbízott személyek által visszaadott értékelésre 0,435-ös alfa értéket kaptunk, vagyis jelentős volt közöttük az egyet nem értés. Ezért azt a 39 (összetett főnév, parafrázist) párt, melynek szórása legalább 1,5 volt, elvetettük. A maradék 111 párra kapott alfa érték 0,696 lett, amit már elfogadhatónak találtunk a feladatra.

A megbízott személyek értékelését úgy használtuk fel, hogy megnéztük azt, hogy átlagosan milyen pontszámot adtak a módszerünk által első, második és harmadik helyen visszaadott parafrázisokra: ezek rendre 3,1842, 2,7687 és 2,5583 voltak. Ez az eredmény azt mutatja, hogy a módszereink által visszaadott parafrázisok átlagban közepesen megfelelőek, és a visszaadott parafrázislistákban előrébb szereplő parafrázisok átlagban jobbak, mint a sorban később szereplő társaik. A feladat nehézségeit figyelembe véve úgy gondoljuk, hogy ezek az eredmények biztatóak, különösen annak fényében, hogy még az angol anyanyelvű értékelők között is nagy az egyet nem értés sok összetett szó értelmezésének tekintetében.

Azt az 5 összetett szót, melyen az algoritmus a legjobb, illetve a legrosszabb eredményt érte el a visszaadott (és nem elvetett) parafrázisok tekintetében, az 1. és 2. táblázatban foglaltuk össze.

5 Konklúzió

Cikkünkben egy olyan módszert mutattunk be, mely alkalmas két főnévből álló angol nyelvű összetett szavak automatikus értelmezésére. Módszerünk először statikus korpuszokban keres az összetett szó értelmezésére alkalmas parafrázisokat, majd webes

kereséseket alkalmazva újrapontozza őket. A módszerünk által első, második és harmadik helyen visszaadott parafrázisokra az anyanyelvi értékelők átlagosan 3,1842, 2,7687 és 2,5583 pontot adtak megfelelőségük alapján (1-től 5-ig terjedő skálán), amit a feladat nehézségeit figyelembe véve biztató eredménynek tartunk.

Mint ahogy azt a 3.2 alfejezetben említettük. idő hiányában nem tudtuk a Web 1T 5-gram Corpust nyelvtanilag elemezni, és a nyelvtani kapcsolatok kinyeréséhez szófaji mintákat használtunk fel. Ez a módszer azonban lényegesen nagyobb hibával jár, mint az automatikus nyelvtani elemzés, ezért a jövőben mindenképpen szeretnénk a már nyelvtanilag elemzett Web 1T 5-gram Corpuson is lefuttatni algoritmusainkat, mely módosítással reményeink szerint eredményeink tovább javulnának. Ezen felül szeretnénk algoritmusainkat további, még nagyobb korpuszok alkalmazásával is kipróbálni, melyek használata szintén kedvezően hathatna az eredményekre.

1. táblázat: Az az 5 összetett szó, melyen az algoritmus a legjobb eredményt érte el.

Összetett főnév, zárójelben a visszaadott parafrázisok	**Átlagos pontszám**
broadway youngster (be in)	4,7500
cell membrane (surround)	4,6000
cattle population (be of)	4,4000
arts museum (be of, be devoted to, be for)	4,3333
business sector (be of)	4,2000

2. táblázat: Az az 5 összetett szó, melyen az algoritmus a legrosszabb eredményt érte el.

Összetett főnév, zárójelben a visszaadott parafrázisok	**Átlagos pontszám**
anode loss (be at, be)	1.5000
bird droppings (be in, be for, be)	1.2667
bow scrape (be)	1.2500
activity spectrum (be in)	1.0000
altitude reconnaissance (-)	1.0000

Hivatkozások

1. Downing, P.: On the creation and use of English compound nouns. Language, Vol. 53 (1977) 810–842
2. Butnariu, C., Kim, S.N., Nakov, P., Seaghdha, D.O., Szpakowicz, S., Veale, T.: Semeval-2010 Task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In: 5th International Workshop on Semantic Evaluation. Taberg Media Group AB, Talberg, Sweden (2009) 100–105
3. Séaghdha, D.O.: Learning compound noun semantics. University of Cambridge, Cambridge, UK (2008)
4. Wright, D.G.S.: Noun-verb associations for Noun-Noun Compound Interpretation. Oxford University Working Papers in Linguistics, Philology & Phonetics, Vol. 8 (2003) 175–190
5. Nakov, P., Hearst, M.: Using Verbs to Characterize Noun-Noun Relations. In: Euzenat, J., Domingue, J. (eds.): Artificial Intelligence: Methodology, Systems, and Applications. Springer, Berlin / Heidelberg, Germany (2006) 233–244

6. Dobó, A., Pulman, S.G.: Interpreting noun compounds using paraphrases. Procesamiento del Lenguaje Natural, Vol. 46 (2011) 59–66
7. Levi, J.N.: The syntax and semantics of complex nominals. Academic Press, New York, USA (1978)
8. Rosario, B., Hearst, M.: Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In: 2001 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg (2001) 82–90
9. Nastase, V., Szpakowicz, S.: Exploring noun-modifier semantic relations. In: 5th International Workshop on Computational Semantics. Association for Computational Linguistics, Stroudsburg (2003) 285–301
10. Lauer, M.: Designing statistical language learners: Experiments on noun compounds. Macquarie University, Sydney, Australia (1995)
11. Nakov, P.: Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics. University of California at Berkeley, Berkeley, USA (2007)
12. Nulty, P., Costello, F.: UCD-PN: Selecting General Paraphrases Using Conditional Probability. In: 5th International Workshop on Semantic Evaluation. Taberg Media Group AB, Talberg, Sweden (2010) 234–237
13. Wubben, S.: UvT: Memory-based pairwise ranking of paraphrasing verbs. In: 5th International Workshop on Semantic Evaluation. Taberg Media Group AB, Talberg, Sweden (2010) 260–263
14. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics, Vol. 16 (1989) 22–29
15. Clark, S., Curran, J.R.: Parsing the WSJ using CCG and log-linear models. In: 42nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg (2004) 103–110
16. Dixon, R.M.W., Aikhenvald, A.U.: Introduction. In: Dixon, R.M.W., Aikhenvald, A.U. (eds.): Changing valency: Case studies in transitivity. Cambridge University Press, Cambridge (2000) 1–29
17. Perlmutter, D.: Impersonal passives and the unaccusative hypothesis. In: 4th Annual Meeting of the Berkeley Linguistics Society. BLS, Berkeley, USA (1978) 157–189
18. Mithun, M.: Valency-changing derivation in Central Alaskan Yup'ik. In: Dixon, R.M.W., Aikhenvald, A.U. (eds.): Changing valency: case studies in transitivity. Cambridge University Press, Cambridge (2000) 84–114
19. Levin, B.: English verb classes and alternations: A preliminary investigation. The University of Chicago Press, Chicago, Il (1993)
20. Kim, S.N., Baldwin, T.: Interpreting noun compounds using bootstrapping and sense collocation. In: 10th Conference of the Pacific Association for Computational Linguistics. Pacific Association for Computational Linguistics, Melbourne, Australia (2007) 129–136
21. Lin, D.: An information-theoretic definition of similarity. In: 15th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1998) 296–304
22. Krippendorff, K.: Content analysis: An introduction to its methodology. Sage Publications, Thousand Oaks, CA, USA (2004)

Félig kompozicionális szerkezetek automatikus felismerése doménadaptációs technikák segítségével a Szeged Korpuszon

Nagy T. István[1], Vincze Veronika[2], Zsibrita János[1]

[1]Szegedi Tudományegyetem, TTIK, Informatikai Tanszékcsoport,
Szeged, Árpád tér 2., e-mail:{nistvan,zsibrita}@inf.u-szeged.hu
[2]Magyar Tudományos Akadémia, Mesterséges Intelligencia Kutatócsoport,
Szeged, Tisza Lajos körút 103., e-mail:vinczev@inf.u-szeged.hu

Kivonat Jelen tanulmányunkban bemutatjuk megközelítésünket, mely félig kompozicionális szerkezeteket képes automatikusan azonosítani magyar nyelvű szövegekben. Első lépésben a lehetséges jelölteket találjuk meg a szövegben, majd egy gazdag jellemzőkészleten alapuló bináris osztályozó segítségével azonosítjuk az egyes félig kompozicionális szerkezeteket. Módszerünket a Szeged Korpusz öt különböző doménjén is megvizsgáljuk, valamint két hasonlósági gráf segítségével azonosítjuk az egymáshoz közel álló részkorpuszokat. A különböző doméneken való vizsgálódások során egy egyszerű doménadaptációs módszert is bemutatunk.

1. Bevezetés

Az olyan főnévből és igéből álló többszavas kifejezéseket, ahol a szemantikai fej a főnév, míg az ige csupán a szerkezet igeiségéért felel, félig kompozicionális szerkezeteknek (FX-ek) nevezzük. Mivel ezen szerkezetek jelentése nem teljesen kompozicionális, ezért azok elemeinek egyenkénti lefordítása nem (vagy csak nagyon ritkán) eredményezi a szerkezet idegen nyelvű megfelelőjét. Az FX-ek automatikus azonosítását továbbá jelentősen megnehezíti, hogy e típusú összetett szerkezetek szintaktikailag hasonló felépítéssel bírnak (*választ kap*), mint más produktív (kompozicionális) szerkezetek (*pulóvert kap*), illetve idiómák (*vérszemet kap*) [1]. Az angol vonzatos igékhez (phrasal verbs) hasonlóan, célszerű az FX-eket is egyetlen komplex egységként kezelni azok nyelvi elemzésekor, hiszen a szerkezet szintaktikai és szemantikai feje nem azonos [2].

Jelen előadásban gépi tanulási megközelítésen alapuló módszerünket ismertetjük, mely magyar nyelven képes a félig kompozicionális szerkezetek automatikus azonosítására folyó szövegben. Továbbá megvizsgáljuk az általunk meghatározott szintaktikai elemzésen alapuló FX-jelöltkiválasztó módszer hatékonyságát. Gépi tanuló megközelítésünk az általunk leírt gazdag jellemzőtéren alapszik, mely egyaránt alkalmaz felszíni jellemzőket, szófaji információkat, funkcióigelistát, valamint szintaktikai és szemantikai információkat.

Módszerünk hatékonyságát a Szeged Korpusz [3] öt különböző doménén (jogi szövegek, fogalmazások, szépirodalmi szövegek, üzleti rövidhírek, újságcikkek) vizsgáltuk meg, melyeken az egyes FX-előfordulások manuálisan annotálva vannak. Mivel úgy találtuk, hogy különböző típusú szövegek különböző típusú félig kompozicionális szerkezeteket tartalmazhatnak, továbbá az FX-ek gyakorisága is eltérhet az egyes doméneken, ezért annak érdekében, hogy ezen különbségeket áthidaljuk, különös figyelmet fordítottunk az egyes korpuszokon tanult modellek hordozhatóságára, melyet egyszerű doménadaptációs technika segítségével valósítottunk meg. Az egyes szövegtípusok közti különbségek bemutatására a különböző doméneken előforduló félig kompozicionális szerkezetek gyakoriságából számított Kendall-együtthatót alkalmaztuk. Ezen domének közti eltéréseket a gépi tanuló algoritmusok által épített modellek által elért eredmények is alátámasztják.

2. Kapcsolódó munkák

Több megközelítést is implementáltak már félig kompozicionális szerkezetek automatikus azonosítására, valamint főnév + ige szerkezetek különböző osztályokba sorolására. Ezek közül a legtöbben alapvetően ige-tárgy párokra koncentráltak, amikor FX-et próbáltak azonosítani. A nem angol nyelvű kutatások során gyakran ige-prepozíció-főnév szerkezeteket vizsgáltak, mint például Van de Cruys és Moirón [4], akik holland nyelvű FX-ek azonosítása során alapvetően szemantikai jellemzőket felhasználó megközelítést alkalmaztak.

Számos megközelítés, mint például Stevenson és társai [5], valamint Van de Cruys és Moirón [4] alapvetően statisztikai jellemzőkre támaszkodva próbált meg automatikusan FX-et azonosítani. Ahogy Vincze [2] is rámutat, egy adott korpuszban az FX-ek nagy többsége igen ritkán fordul elő egy adott korpuszon. A vizsgált nagyméretű szövegeken az FX-ek 87%-a fordul elő kevesebb mint háromszor, ennél fogva igen nehéz pusztán statisztikai jellemzők alapján azonosítani őket.

Diab és Bhutada [6], valamint Nagy T. és társai [7] jellemzően (sekély) nyelvi információkra támaszkodó szabályalapú rendszereket alkalmaztak FX-ek azonosítására. Vincze és társai [8] szabályalapú rendszerüket mind magyar, mind angol nyelven alkalmazták többek közt a SzegedParallelFX párhuzamos korpuszon.

Statisztikai és nyelvi információkat egyaránt felhasználó rendszert építettek többek közt Tan és társai [9], valamint Tu és Roth [10]. Mindkét megközelítés ige + főnév párokat osztályoz aszerint, hogy félig kompozionális szerkezet-e vagy sem. Tu és Roth mind környezeti, mind statisztikai jellemzőket felhasználva tanított egy támasztóvektorgép-modellt a pozitív és negatív példák számában kiegyensúlyozott adathalmazon. Tanulmányuk szerint a többértelmű példákon a lokális jellemzőket használva érhetünk el jobb eredményeket. A Tan és társai által alkalmazott gépi tanuló alkalmazás statisztikai, valamint nyelvi információkat kombinálva véletlen erdő módszerét alkalmazva osztályozta a lehetséges FX-jelölteket.

Az általunk megvalósított megközelítés szintaktikai jellemzők alapján automatikusan kinyert főnév + ige párokat osztályoz gazdag jellemzőtérre támaszkodó gépi tanuló módszer alapján.

3. A félig kompozicionális szerkezetek automatikus azonosítása

Jelen munkában elsődleges célunk minden félig kompozicionális szerkezet automatikus azonosítása magyar nyelvű folyó szövegekben.

Mivel a különböző típusú szövegek merőben eltérő félig kompozicionális szerkezeteket tartalmazhatnak, valamint a különböző szövegekben más-más arányban fordulhatnak elő ezen szerkezetek, ezért fontosnak találtuk megvizsgálni az egyes doménen tanult modellek hordozhatóságát. Ezért módszereink kiértékelésére a Szeged Korpuszt használtuk, melyen öt különböző típusú szövegben vannak a félig kompozicionális szerkezetek manuálisan annotálva. Habár a korpuszban az FX-ek melléknévi igenévi és főnévi alakjai is jelölve vannak, mi alapvetően csak az igei alakok felismerésére fókuszáltunk. A Szeged Korpusz adatai az 1. táblázatban találhatók.

1. táblázat. A Szeged Korpusz adatai.

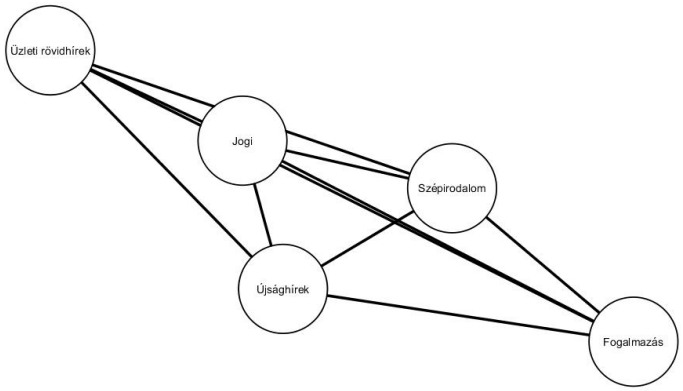
Korpusz	**Mondatok száma**	**Tokenek száma**	**FX**
Fogalmazás	23136	314787	677
Jogi	7058	188899	698
Szépirodalom	17358	219784	634
Üzleti rövidhírek	8956	213936	582
Újsághírek	8848	191156	484
Összesen	65356	1128562	3075

Mivel az alkalmazott megközelítésünk nagymértékben támaszkodik a szintaktikai jellemzőkre, ezért a Szeged Korpusznak csak azon részét használtuk fel, melyre a `magyarlanc 2.0` [11] szintaktikai elemzést tudott adni. Így végül öt különböző doménen 65356 mondaton 3075 FX-et vizsgáltunk. Az egyes részkorpuszokon tízszeres keresztvalidációval tanított és predikált modellek szófaji és függőségi elemzését használtuk. Mivel az etalon szófaji és függőségi elemzések egyaránt elérhetőek a Szeged Korpuszon, ezért lehetőségünk nyílt megvizsgálni, milyen hatással vannak a `magyarlanc 2.0` által nyújtott automatikus nyelvi elemzések megközelítésünk eredményességére. A különböző domének összehasonlítására kiszámoltuk az egyes részkorpuszokon a 15 leggyakrabban előforduló félig kompozicionális szerkezet Kendall-konkordancia értékeit, melyek a 2. táblázatban láthatóak.

A Kendall-együttthatók értékei alapján az egyes részkorpuszok hasonlóságát a 1. ábrán látható doménhasonlósági gráf segítségével ábrázoltuk, ahol az FX-

2. táblázat. Részkorpuszok Kendall-konkordancia értékei a 15 leggyakrabban előforduló félig kompozicionális szerkezet alapján.

-	**Fogalmazás**	**Jogi**	**Szépirodalom**	**Üzleti rövidhírek**	**Újsághírek** -
Fogalmazás	1	0,1825	0,5883	0,064	0,2498
Jogi	0,1825	1	0,2849	0,5068	0,3922
Szépirodalom	0,5883	0,2849	1	0,2422	0,2417
Üzleti rövidhírek	0,064	0,5069	0,2422	1	0,2409
Újsághírek	0,2498	0,3922	0,2417	0,2409	1

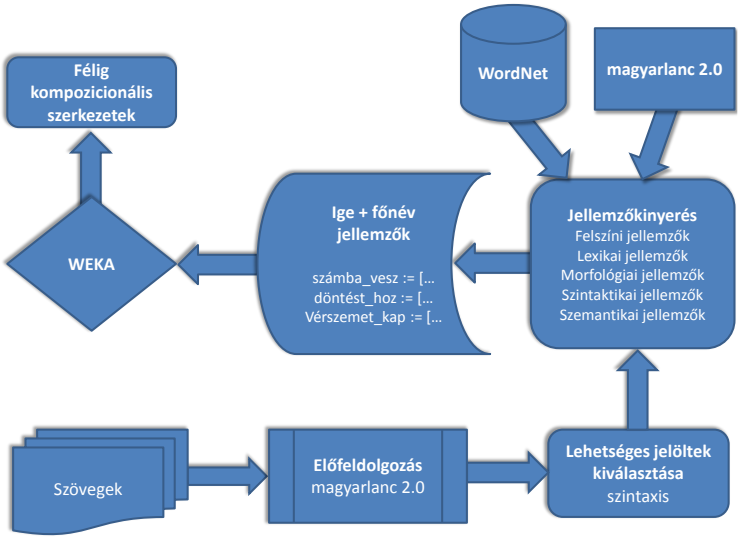


1. ábra. Doménhasonlósági gráf Kendall-együttható alapján.

ek szempontjából hasonló típusú szövegek közelebb, míg a kevésbé hasonlóak távolabb helyezkednek el egymástól.

3.1. Gépi tanuló megközelítés félig kompozicionális szerkezetek automatikus azonosítására

A félig kompozicionális szerkezetek automatikus azonosítására egy gépi tanuló megközelítést implementáltunk. Ehhez első lépésben minden mondatot elemzünk, és a lehetséges félig kompozicionális szerkezeteket szintaxisalapú jelöltkiválasztó megközelítés segítségével automatikusan kinyerjük. A második lépésben egy gazdag jellemzőkészleten alapuló bináris osztályozó segítségével döntünk, hogy egy adott potenciális szerkezet valóban félig kompozicionális szerkezet-e vagy sem. A 2. ábra mutatja be a teljes rendszer működését.



2. ábra. Rendszerábra.

3.2. Automatikus jelöltkinyerés

Azáltal, hogy az egyes félig kompozicionális szerkezetek a Szeged Korpusz részkorpuszain manuálisan annotálva vannak, lehetőségünk nyílt megvizsgálni ezen szerkezetek szintaktikai kapcsolatait folyó szövegekben. Ezen vizsgálataink alapján a lehetséges félig kompozicionális szerkezetekre úgy tekintettünk, mint olyan ige-főnév párok, melyek közt `subj`, `obj`, vagy `obl` (alany, tárgy vagy egyéb argumentum) szintaktikai kapcsolat van. Ahogy a 3. táblázatban látható, ezzel a jelöltkinyerő megközelítéssel képesek vagyunk a félig kompozicionális szerkezetek 92,07%-át automatikusan azonosítani.

3. táblázat. Az egyes részkorpuszokon előforduló félig kompozicionális szerkezetek szintaktikai kapcsolatai.

Korpusz	**OBJ**	**OBL**	**SUBJ**	**Összesen**	**Etalon**	**Fedés %**
Fogalmazás	401	171	45	617	677	91,14%
Jogi	394	150	97	641	698	91,83%
Szépirodalom	296	257	27	580	634	91,48%
Üzleti rövidhírek	339	176	19	534	582	91,75%
Újsághírek	307	130	22	459	484	94,83%
Összesen	1737	884	210	2831	3075	92,07%

3.3. Gépi tanuló alapú automatikus jelöltosztályozás

A következőkben bemutatjuk gépi tanuló alapú megközelítésünket, amelyet a lehetséges félig kompozionális szerkezetek automatikus osztályozására implementáltunk, és amely a következő osztályokba sorolható gazdag jellemzőkészleten alapszik: felszíni, lexikai, morfológiai, szintaktikai és szemantikai.

- Felszíni jellemzők: a **végződés** jellemző azt vizsgálja, hogy a szerkezet főnévi tagja bizonyos bi- vagy trigramra végződik-e. Ezen jellemző alapja, hogy az FX-ek főnévi komponense igen gyakran egy igéből képzett főnév. A szerkezetet alkotó **tokenek száma** szintén jellemzőként lett felhasználva.
- Lexikai jellemzők: A **leggyakoribb ige** jellemző az FX-ek azon tulajdonságát használja fel, hogy általában a leggyakoribb igék szerepelnek funkcióigeként (például *ad*, *vesz*, *hoz* stb.). Ezért az FX-jelöltek igei komponensének lemmáját vizsgáltuk, hogy az megegyezik-e az előre megadott leggyakoribb igék egyikével. A SzegedParalellFX korpuszban manuális annotált FX-ből gyűjtött, lemmatizált **FX lista** is felhasználásra került mint bináris jellemző, amely akkor kapott igaz értéket, ha az adott potenciális FX szerepelt a listában.
- Morfológiai jellemzők: mivel a magyar nyelv igen gazdag morfológiával rendelkezik, ezért számos morfológiaalapú jellemzőt definiáltunk. A **POS** módszernél FX-ekre jellemző szófaji mintákat definiáltunk, és amennyiben az FX-jelöltre illeszkedett egy minta, a jellemző igaz értéket kapott. További jellemzőként definiáltuk a funkcióigék **MSD-kódját** felhasználva az ige módját (`Mood`), valamint a főnévi komponens típusát (`SubPos`), esetét (`Cas`), a birtokos számát (`NumP`), a birtokos személyét (`PerP`), valamint a birtok(olt) számát (`NumPd`). A **szótő** jellemző alapvetően a főnévi komponens szótövét vizsgálja. Ez a jellemző az FX-ek azon már említett tulajdonságát kívánja kihasználni, hogy a félig kompozicionális szerkezetek főnévi tagja igen gyakran egy igéből származik, ezért azt vizsgáltuk, hogy a főnév tag szótövének van-e igei elemzése.
- Szintaktikai jellemzők: potenciális FX-ek kiválasztásánál alapvetően **szintaktikai információkra** támaszkodtunk. Ugyanakkor jellemzőként definiáltuk, hogy a három szintaktikai osztály (alany, tárgy vagy egyéb) melyike áll fenn az aktuális FX-jelölt esetében.
- Szemantikai jellemzők: ebben az esetben is az FX azon tulajdonságát használtuk fel, hogy a főnévi tag igen gyakran egy igéből származik. Ezért a Magyar WordNet-et [12] felhasználva `tevékenység` vagy `esemény` **szemantikai jelentést** keresünk a főnévi tag felsőbb szintű hipernimái közt.

Mivel a fentebb ismertetett jellemzők nagy része bináris attribútum, ezért a WEKA [13] csomagban elérhető, a C4.5 [14] döntési fa algoritmust implementáló J48 tanuló algoritmust alkalmaztuk. Rendszerünket minden részkorpuszon mondatszintű tízszeres keresztvalidációval értékeltük ki. A kiértékelés során a pontosság, fedés és F-mérték metrikákat használtunk. Ahogy a 3. táblázatban is látható, a potenciális FX-jelölt kiválasztó megközelítésünk az egyes korpuszokban manuálisan annotált FX-k 92,07%-át fedi csak le, ezért a gépi tanuló megközelítések fedés eredményeit korrigálnunk kellett.

Az egyes részkorpuszok összehasonlítására egyszerű, domének közötti keresztméréseket alkalmaztunk, mely során a forráskorpuszon tanított modelleket értékeltük ki a célkorpuszokon. Tehát a tanítóhalmaz nem tartalmazott annotált mondatokat a célkorpuszról.

Amennyiben nagyobb számú etalon példa áll rendelkezésünkre más-más doménekről és csak korlátozott számú példával rendelkezünk a feladat szempontjából érdekes doménről, akkor doménadaptációs technikák segítségével javíthatjuk rendszerünk hatékonyságát. Vagyis hatékonyabb gépi tanuló modellt építhetünk, ha a nagyméretű forrásdomén tanítóhalmazt kiegészítjük a céldoménen elérhető kisebb etalon korpusszal.

A Szeged Korpusz öt különböző típusú részkorpuszának köszönhetően megvizsgálhattuk, hogy egyszerű doménadaptációs technikák segítségével hogyan növelhetjük rendszerünk teljesítményét. Egy nagyon egyszerű doménadaptációs megoldást alkalmaztunk: a tanítóhalmazt kiegészítettük 500 céldoménről véletlenszerűen kiválasztott mondattal, majd 500 mondatonként növeltük a céldoménről érkező mondatok számát egészen 3000-ig. A doménadaptáció kiértékelésére is mondatszintű tízszeres keresztvalidációt alkalmaztunk. Az eredmények összehasonlíthatósága érdekében a keresztvalidáció során ugyanazon teszthalmazokat alkalmaztuk a céldoménen, mint a doménen belüli kiértékelés során. Ugyanakkor figyelmet fordítottunk arra is, hogy a doménadaptációhoz véletlenszerűen kiválasztott mondatok egyike se szerepeljen az aktuális teszthalmazban.

Baseline megoldásnak szótárillesztési megközelítést vettünk. Minden részkorpusz esetében a gépi tanuló megközelítésben is alkalmazott, a SzegedParallelFX korpuszon manuálisan annotált FX-ekből létrehozott lista lemmatizált verzióját használtuk a szótárillesztés során. Amennyiben a lista egy eleme előfordult egy adott mondat lemmatizált verziójában, akkor azt FX-nek jelöltük. Az etalon, valamint predikált jellemzőket felhasznált gépi tanult modellek eredményei és a szótárillesztés eredményei a 4. táblázatban, míg a keresztmérések eredményei a 6. táblázatban találhatók.

4. Eredmények

A tízszeres keresztvalidációval kiértékelt eredmények alapján a jogi korpuszon értük el a legjobb eredményeket 68,35 F-mértékkel. Ugyanakkor a legnehezebb doménnek a fogalmazás (51,83 F-mérték) és az újsághírek (51,84 F-mérték) részkorpuszok bizonyultak. Az etalon és predikált jellemzőkön tanult gépi tanuló modellek közt a szépirodalmi korpuszon volt a legnagyobb, 1,5 pontos eltérés, míg az üzleti rövidhírek esetében csupán 0,23 pontos különbség mutatkozott. Az öt korpuszon átlagosan 0,69 ponttal bizonyultak jobbnak az etalon jellemzőket használó modellek a predikált jellemzőket használóknál. A szótárillesztés a fogalmazás doménen bizonyult a leghatékonyabbnak 32,91 pontos F-mértékkel, és szintén ezen a részkorpuszon mutatkozott a legkisebb eltérés a gépi tanuló modell és baseline megközelítés közt. Szemben a jogi doménnel, ahol a két megközelítés közt 41,76 pontos eltérés mutatkozott.

4. táblázat. Szótárillesztés, valamint a gépi tanult megközelítés eredményei a különböző doméneken, etalon és predikált jellemzőket felhasználva.

Korpusz	**Pontosság**	**Fedés**	**F-mérték**	**Különbség**
Fogalmazás				
etalon	53,05	50,66	51,83	-
predikált	54,18	48,74	51,32	-0,51
szótárillesztés	52,85	23,88	32,91	-18,92
Jogi				
etalon	68,65	68,05	68,35	-
predikált	68	66,91	67,45	-0,9
szótárillesztés	47,52	18,46	26,59	-41,76
Szépirodalom				
etalon	56,72	47,48	51,69	-
predikált	52,27	48,26	50,19	-1,5
szótárillesztés	68,81	23,71	35,26	-16,43
Üzleti rövidhírek				
etalon	65,04	57,9	61,26	-
predikált	62,51	59,62	61,03	-0,23
szótárillesztés	53,48	18,42	27,39	-33,87
Újsághírek				
etalon	49,56	54,34	51,84	-
predikált	51,17	51,86	51,51	-0,33
szótárillesztés	43,72	20,52	27,93	-23,91
Átlag				
etalon	49,56	54,34	56,99	-
predikált	57,63	55,08	56,3	-0,69
szótárillesztés	53,28	20,99	30,02	-26,97

5. táblázat. Az egyes jellemzőosztályok.

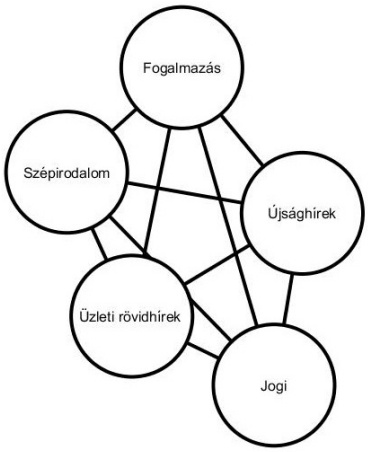
Jellemző	**Pontosság**	**Fedés**	**F-mérték**	**Eltérés**
Felszíni	53,73	56,19	54,93	-6,1
Lexikai	47,98	40,38	43,85	-17,18
Morfológiai	61,34	57,56	59,39	-1,64
Szintaktikai	61,35	59,11	60,21	-0,82
Szemantikai	63,4	56,76	59,9	-1,13
Összes	62,51	59,62	61,03	0

Hogy megvizsgálhassuk, az egyes jellemzők miként befolyásolják a gépi tanuló rendszer eredményeit, az üzleti rövidhír részkorpuszon porlasztásos mérést végeztünk, melynek eredményei a 5. táblázatban láthatók. Ekkor a teljes jellemzőtérből elhagytuk az egyes jellemzőcsoportokat, majd a maradék jellemzőkre támaszkodva tanítottunk. Az eredmények alapján a leghasznosabbnak a lexikai, valamint a felszíni jellemzők bizonyultak. A lexikai jellemzők közül elsősorban a funkcióige-lista bizonyult a leghatékonyabb jellemzőnek.

A keresztmérések alapján, a fogalmazás korpuszon a szépirodalmi doménen tanított modell teljesített a legjobban 43,29 pontos F-mértéket elérve. Ugyan 11,96 ponttal kisebb F-mértéket tudott elérni az üzleti rövidhíreken tanult modell a jogi részkorpuszon a céldoménhez képest, ám így is ez a modell volt a leghatékonyabb a többi közül. A szépirodalmi doménen a fogalmazás korpuszon tanult megközelítése bizonyult a legjobbnak 49,84 pontos F-mértékkel. Üzleti rövidhírek esetében a legjobb eredményt az újsághíreken tanított gépi tanulási modell érte el 55,75 pontos F-mértékkel. 50,42 pontos F-mértékkel az üzleti rövidhíreken tanított, ám az újsághíreken predikált modell bizonyult a legjobbnak.

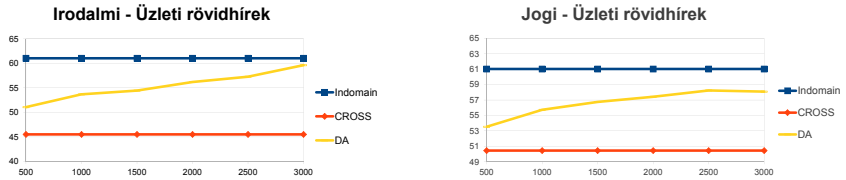
6. táblázat. Keresztmérések eredményei az egyes részkorpuszokon.

Korpusz	**Pontosság**	**Fedés**	**F-mérték**	**Eltérés**
Fogalmazás	54,18	48,74	51,32	-
Jogi	20,08	39,44	26,61	-24,71
Szépirodalom	37,62	50,96	**43,29**	-8,03
Üzleti rövidhírek	37,31	36,93	37,12	-14,02
Újsághírek	37,62	29,39	33	-18,32
Jogi	68	66,91	67,45	-
Szépirodalom	52,98	47,13	49,89	-17,56
Fogalmazás	55,21	40,26	46,56	-20,89
Üzleti rövidhírek	64,22	48,85	**55,49**	-11,96
Újsághírek	69,18	42,12	52,36	-15,09
Szépirodalom	52,27	48,26	50,19	-
Jogi	27,92	32,81	30,17	-20,02
Fogalmazás	60,75	42,19	**49,84**	-0,35
Üzleti rövidhírek	51,04	38,64	43,99	-6,2
Újsághírek	42,04	20,82	27,85	-22,34
Üzleti rövidhírek	62,51	59,62	61,03	-
Jogi	43,89	59,28	50,44	-10,59
Szépirodalom	40,85	51,37	45,51	-15,52
Fogalmazás	48,22	34,88	40,48	-20,55
Újsághírek	60	52,06	**55,75**	-5,28
Újsághírek	51,17	51,86	51,51	-
Jogi	30,76	61,78	41,07	-10,44
Szépirodalom	34,8	55,58	42,8	-8,71
Fogalmazás	40,64	41,74	41,18	-10,33
Üzleti rövidhírek	46,29	55,37	**50,42**	-1,09



3. ábra. Doménhasonlósági gráf keresztmérések eredményei alapján.

A keresztmérések eredményei alapján az egyes domének közti hasonlóságokat a 3. ábrán látható irányítatlan, súlyozott gráf segítségével jelenítettük meg. A gráf súlyait az adott domén tízszeres keresztvalidációval mért eredményei, valamint a keresztmérések különbségei adták.



4. ábra. Doménadaptációs eredmények üzleti rövidhírek doménen, irodalmi és jogi részkorpuszon tanítva.

A doménadaptációs mérések eredményei a 4. ábrán látható. A két kép bemutatja, hogy az adaptációhoz használt mondatok számának változásával hogyan módosul az adott doménen a rendszer által elért F-mérték.

Mind a két esetben jól látszik, hogy az adaptációhoz a céldoménről felhasznált mondatok számával folyamatosan növekednek a céldoménen elért eredmények. Az irodalmi részkorpuszt forrásdoménként használva, a doménadaptáció segítségével a céldoménen tízszeres keresztvalidációval elérhető eredményét közelítettük

meg. A doménadaptáció határozottan képes volt javítani a jogi részkorpusz forrásdoménről történő keresztmérés eredményéhez képest.

5. Az eredmények értékelése, összegzés

Jelen munkánkban bemutattuk gazdag jellemzőtéren alapuló gépi tanuló megközelítésünket, mely automatikusan képes magyar nyelvű szövegekben félig kompozicionális szerkezeteket azonosítani. A problémát két lépésből álló megközelítéssel oldottuk meg: az első lépésben a folyó szöveg mondataiból a potenciális FX-jelölteket nyertük ki automatikusan, egy alapvetően szintaxisra támaszkodó jelöltkiválasztó megközelítéssel. Módszerünk igen hatékonynak bizonyult, mivel a manuálisan annotált FX-ek 92%-át sikerült lefedje. A kinyert példák közül automatikusan azonosítottuk az egyes FX-eket egy gazdag jellemzőtéren alapuló bináris osztályozó segítségével. Módszerünket a Szeged Korpusz egyes doménjein értékeltük ki, azt vizsgálva, mely részkorpuszok hasonlítanak a leginkább egymásra, melyeken fordulnak elő hasonló FX-ek.

Az egyes domének közötti hasonlóságok kifejezésére két hasonlósági gráfot is megadtunk. Az első esetben az egyes részkorpuszokon előforduló FX-ek gyakoriságából számított Kendall-együtthatóval súlyoztuk a gráf egyes éleit, míg a másik esetben a keresztmérések eredményei alapján lettek a gráf élei súlyozva. Ezek alapján megállapítható, hogy a fogalmazás és a szépirodalom domének, valamint a újsághírek és üzleti hírek domének hasonlítanak egymásra a legjobban. A jogi szövegek pedig inkább az utóbbi két részkorpuszhoz hasonlítanak.

Rendszerünk hibaelemzése is alátámasztotta a porlasztásos mérés során is bemutatott eredményt, miszerint a leghatékonyabb jellemzőnek a funkcióige-lista bizonyult. Ugyanis a hibaelemzés során kiderült, hogy a helyesen predikált FX-ek igéinek több mint 80%-a szerepelt a funkcióige-listában, míg az álpozitív FX-ek igéinek kevesebb mint 10% volt megtalálható a listában. Az elemzés arra is enged következtetni, hogy rendszerünk alapvetően a rövidebb, kevesebb mint 3 tokenből álló FX-t azonosítja helyesen. Továbbá néhány álpozitív eredmény annotálási hibára, valamint helytelen szófajkódi elemzésre vezethető vissza.

Megközelítésünket különböző doméneken is kiértékeltük, az egyes részkorpuszokon elérhető eredményeket pedig egyszerű doménadaptációs technikákkal javítottuk. Eredményeink azt mutatják, hogy a magyar nyelvű FX-ek folyó szövegben való automatikus azonosítása igen kihívásokkal teli feladat, de az általunk bemutatott megközelítés erre a nehéz problémára nyújt egy lehetséges megoldást.

Köszönetnyilvánítás

Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

1. Vincze, V.: Light Verb Constructions in the SzegedParalellFX English–Hungarian Parallel Corpus. In: Proceedings of LREC-2012, Istanbul, Turkey, ELRA (2012) 2381–2388
2. Vincze, V.: Semi-Compositional Noun + Verb Constructions: Theoretical Questions and Computational Linguistic Analyses. Doktori értekezés, Szeged, Szegedi Tudományegyetem (2011)
3. Alexin, Z., Gyimóthy, T., Hatvani, Cs., Tihanyi, L., Csirik, J., Bibok, K., Prószéky, G.: Manually annotated Hungarian corpus. In: Proceedings of EACL-2003 - Volume 2. EACL '03, Stroudsburg, PA, USA, ACL (2003) 53–56
4. Van de Cruys, T., Moirón, B.n.V.: Semantics-based multiword expression extraction. In: Proceedings of the Workshop on a Broader Perspective on Multiword Expressions. MWE '07, Stroudsburg, PA, USA, ACL (2007) 25–32
5. Stevenson, S., Fazly, A., North, R.: Statistical measures of the semi-productivity of light verb constructions. In: Proceedings of the Workshop on Multiword Expressions: Integrating Processing. MWE '04, Stroudsburg, PA, USA, ACL (2004) 1–8
6. Diab, M.T., Bhutada, P.: Verb noun construction MWE token supervised classification. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications. MWE '09, Stroudsburg, PA, USA, ACL (2009) 17–22
7. Nagy T., I., Vincze, V., Berend, G.: Domain-Dependent Identification of Multiword Expressions. In Angelova, G., Bontcheva, K., Mitkov, R., Nicolov, N., eds.: RANLP, RANLP 2011 Organising Committee (2011) 622–627
8. Vincze, V., Nagy T., I., Zsibrita, J.: Félig kompozicionális szerkezetek automatikus azonosítása magyar és angol nyelven. In Tanács, A., Vincze, V., eds.: VIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem (2011) 59–70
9. Tan, Y.F., Kan, M.Y., Cui, H.: Extending corpus-based identification of light verb constructions using a supervised learning framework. In: Proceedings of the EACL Workshop on Multi-Word Expressions in a Multilingual Contexts, Trento, Italy, ACL (2006) 49–56
10. Tu, Y., Roth, D.: Learning English Light Verb Constructions: Contextual or Statistical. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World, Portland, Oregon, USA, ACL (2011) 31–39
11. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In Tanács, A., Vincze, V., eds.: MSzNy 2013 – IX. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem (2013) 368–374
12. Miháltz, M., Hatvani, Cs., Kuti, J., Szarvas, Gy., Csirik, J., Prószéky, G., Váradi, T.: Methods and Results of the Hungarian WordNet Project. In Tanács, A., Csendes, D., Vincze, V., Fellbaum, C., Vossen, P., eds.: Proceedings of the Fourth Global WordNet Conference (GWC 2008), Szeged, University of Szeged (2008) 311–320
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations **11**(1) (2009) 10–18
14. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)

Automatikusan generált online szótárak: az EFNILEX projekt eredményei

Héja Enikő, Takács Dávid

MTA Nyelvtudományi Intézet
1068 Budapest, Benczúr u. 33.
{eheja,takdavid}@nytud.hu

Kivonat: Az előadás összefoglalja a 2008-ban kezdődő EFNILEX lexikográfiai projekt munkálatait, különös tekintettel a 2012-ben elért eredményekre. A projekt célja annak a vizsgálata volt, hogy a nyelvtechnológiai eszközök és eljárások mennyiben alkalmasak a kétnyelvű szótárak előállításának támogatására. Ennek elsősorban a kevéssé használt nyelvek esetében van jelentősége, hiszen ezeket a szótárakat a kereskedelmi kiadók nem tartják piacképesnek, így az elkészítésükbe sem invesztálnak jelentősen.

Mivel még nem léteznek olyan módszerek, amelyek a szótárak teljesen automatikus előállítását lehetővé teszik, eredeti célkitűzésünk az volt, hogy a lexikográfusokat olyan automatikusan generált erőforrásokkal lássuk el, amely a lehető legjobban csökkentik a szótárak elkészítéséhez szükséges munkát. Ezeket az erőforrásokat protoszótáraknak nevezzük. Természetesen annak a lehetőségét sem zártuk ki, hogy ezek az erőforrások valamilyen formában közvetlenül is számot tarthatnak a szótárhasználók érdeklődésére.

A protoszótárak előállítási folyamatának lényege, hogy a fordítási ekvivalenciákat szóillesztés útján nyerjük ki kétnyelvű párhuzamos korpuszokból. A módszer egyik előnye, hogy a lexikai megfeleltetések kiválasztása korpuszvezérelt módon történik, amely által a lexikográfusi intuíció csökkenthető. További előny, hogy az egyes fordításokhoz kétnyelvű konkordanciák állnak rendelkezésre, amelyek segítséget nyújtanak az egyes fordítások használati feltételeinek karakterizálásában. Ezen felül, nézetünk szerint ez a fajta megközelítés jobban illeszkedik a szótárhasználók igényeihez, hiszen szótárhasználatkor az elsődleges cél általában szövegek, és nem elszigetelt szavak megértése, illetve létrehozása (pl. [3]).

A protoszótárak újdonsága a hagyományos szótárakkal szemben, hogy a tartalmuk automatikusan testre szabható bizonyos mérőszámok mentén, amelyeket a statisztikai szóillesztés során határozunk meg. Az így testre szabott szótárak egy megfelelő lekérdező felülettel ellátva pedig a szótárhasználók számára már közvetlenül is hasznosak lehetnek. Az online szótárak lekérdezhetőek a http://efnilex.efnil.org oldalon (pl. [4]).

Az eredményül kapott szótárak a módszer lényegéből fakadóan számos újdonságot tartalmaznak a hagyományos szótárakkal szemben. Először is, a szótárak testreszabhatósága lehetővé teszi, hogy a szótárak különböző felhasználói szinteknek feleljenek meg. Így például a megfelelő paraméterek beállításával a felhasználó kiválaszthat egy

olyan szótárt, amely csak a leggyakoribb forrásnyelvi szavakat és ezek legvalószínűbb fordításait tartalmazza. Egy ilyen szótár tökéletes egy kezdő nyelvtanuló számára. Egy további lehetőség, hogy a felhasználó egy viszonylag nagy lefedettségű szótárt vág ki és kérdez le, amely már tartalmazhat hibás fordítási jelölteket is. Egy ilyen szótár a professzionális felhasználók, pl. fordítók számára lehet érdekes, akiket sokszor a speciális fordítási lehetőségek érdekelnek, ugyanakkor rendelkeznek kellő nyelvismerettel ahhoz, hogy az esetleges téves fordítási jelölteket kiszűrjék.

A felhasználói felület további előnyei közé tartozik, hogy a nyelvek közötti szemantikai relációkra is javaslatot tesz. Ez azért is nagyon fontos, mert a szigorú értelemben vett fordítási ekvivalencia – amikor a forrás- és célnyelvi kifejezés pontosan ugyanolyan kontextusokban jelennek meg – ritka jelenség (pl. [1]). Így fontos, hogy a szótár tartalmazzon arra vonatkozó javaslatokat, hogy a célnyelvi kifejezés használata megszorítottabb vagy általánosabb-e, mint a forrásnyelvi kifejezésé (pl. a magyar *tisztán* szó egyaránt fordítható *clearly*-nek és *distinctly*-nek az angolban, de hasznos, ha a szótár jelzi, hogy az utóbbi fordítás megszorítottabb környezetekben fordulhat csak elő).

Mindazonáltal ezek az újdonságok még nem teljes mértékben kidolgozottak, a paraméterbeállítások még további pontosításra szorulnak. Ezenfelül a felhasználói felületet is célunk felhasználóbarátabbá tenni. Részben ezen célokat szolgálja, hogy elkészítettük a Hunglish 1.0-n [5] alapuló angol-magyar, magyar-angol protoszótárainkat is, amelyek lekérdezhetővé tételük után reményeink szerint segítenek eredményeink disszeminálásában, valamint a felhasználói javaslatok alapján a további fejlesztések pontos meghatározásában is.

A módszer hátrányai közé tartozik, hogy a megfelelő méretű párhuzamos korpusz összegyűjtése főleg a kevéssé használt nyelvek esetében nehézkes. További hátrány, hogy a módszer önmagában nem kezeli a többszavas kifejezéseket.

A projekt 2012-es szakaszában a többszavas kifejezések kinyerésére is koncentráltunk, ezen belül is a többnyelvű kollokációk kinyerésére. Többnyelvű kollokációkat a következő nyelvpárokra vontunk ki: magyar-szlovén, magyar-litván, illetve magyar-angol. A munkafolyamat 3 lépésből áll. (1) Minden nyelvre külön-külön kinyerjük az egynyelvű kollokációkat. (2) A kinyert kollokációkat felismerjük a párhuzamos korpuszok releváns részében és egytokenes kifejezéssé alakítjuk, hogy ezek is a szóillesztő algoritmus bemenetéül szolgálhassanak. (3) A szóillesztő algoritmust futtatva nyerjük ki a kollokációkat és a hozzájuk tartozó fordítási jelölteket.

Az egynyelvű kollokációk kinyerése során ezúttal csak szomszédos tokeneket vettünk figyelembe, amelyekre szófaji megkötést is tettünk. A magyar-szlovén, a magyar-litván és a magyar-angol nyelvpárok esetében az AN, AdvV, NN formájú kollokációkat vettük figyelembe, ahol A jelöli a mellékneveket, N a főneveket, Adv a határozószókat és V az igéket. Az angol-magyar esetében további kollokációtípusokat is figyelembe vettünk: a magyar oldalon az NV formájú kollokációkat, melyek az igemódosítós igéket tartalmazzák, angol oldalon pedig a VN formájú kollokációkat, amely kategória, hipotézisünk szerint, elsősorban ige+névelőtlen tárgy szerkezeteket tartalmaz. Az előbbiekkel szemben ez a két szerkezet nem teljesen párhuzamos egymással szintaktikailag, hiszen a magyarban igemódosító pozícióban nemcsak tárgyesetű főnevek jelenhetnek meg (pl. *iskolába jár*). A névelőtlenség miatt

mégis azt gondoltuk, hogy ez a kategória felel meg legjobban az angol VN szerkezeteknek.

A kollokációk kinyerésére az UCS 0.6 szabadon elérhető kollokációkinyerő eszközt használtuk [2]. A kollokációjelöltek az ötnél nem ritkábban előforduló szópárok voltak, amelyek a fent említett formai kritériumoknak megfeleltek. A következő lépésben ezeket két különböző asszociációs mérték szerint szűrtünk: MI és Z-score szerint. A szóillesztő futtatása után azokat a fordítási párokat vettük figyelembe, amelyeknek vagy a célnyelvi vagy a forrásnyelvi oldalán kollokáció szerepelt. Az eredményül kapott fordítási jelölteket lekérdezhetővé tettük. Így az automatikusan generált szótárakból kiderül, hogy az *arc* lehet *beesett*, *eltorzult*, *kipirult*, *sápadt*, és a *beesett arc* egy lehetséges angol fordítása: *hollow cheek*. Az eredmények részletes kiértékelése a közeljövő feladata.

A párhuzamos kollokációk kinyerésének megkönnyítésére egy fontos fejlesztést vezettünk be: minden rendelkezésre álló párhuzamos korpuszt (Hunglish 1.0, litván-magyar, szlovén-magyar) egységes XML-annotációval láttunk el. Ez kettős célt szolgál: (1) a vizsgálni kívánt szerkezeti egységek kinyerését lényegesen megkönnyíti; (2) a kollokációk egységes kezeléséhez célszerűnek tűnt egy (kvázi) egységes morfoszintaktikai annotáció bevezetése. Az így újragenerált szótárak a kollokációkon túl szófaji információt is tartalmaznak, sőt bizonyos esetekben megadják azt is, hogy egy tipikus fordítás jellemzően milyen típusú szövegekben fordul elő.

Hivatkozások

1. Atkins, B.T. S., Rundell, M.: The Oxford Guide to Practical Lexicography. Oxford University Press (2008)
2. Evert, S.: The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, URN urn:nbn:de:bsz:93-opus-23714 (2005)
3. Héja, E.: The Role of Parallel Corpora in Bilingual Lexicography. In: Proceedings of the LREC2010 Conference. La Valletta, Malta, May 2010 (2010) 2798–2805
4. Héja, E., Takács D.: Automatically Generated Customizable Online Dictionaries. In: Daelemans W. et al. (eds.): Proceedings of EACL2012. The Association for Computational Linguistics, Avignon, France (2012) 51–57
5. Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy V.: Parallel corpora for medium density languages. In: Proceedings of the RANLP 2005 (2005) 590–596

A 4lang fogalmi szótár

Kornai András, Makrai Márton

MTA SZTAKI
Nyelvtechnológiai Kutatócsoport
e-mail: kornai@sztaki.hu

1. Bevezetés

A 4lang fogalmi szótár három, a számítógépes nyelvészeti és pszicholingvisztikai alkalmazások számára fontos célt szolgál. Egyrészt feladata az *alapszókincs* meghatározása, ezzel a cikk első részében foglalkozunk. Másrészt feladata a *definíciós* tevékenység támogatása, ezt a cikk második, szótárunkat a lexikográfia elméleti keretei közt elhelyező része tárgyalja. Végül célunknak tekintjük a szövegmegértési feladatok (kérdésmegválaszolás, információ-visszakeresés, gépi fordítás) támogatását is, ezzel a cikk harmadik, kitekintő része foglalkozik.

Konceptuális szótárunk lemmái többnyelvűek. Jelenleg a magyaron kívül angol, latin és lengyel, innen a 4lang elnevezés, de hosszabb távon ennek 40 nyelvre való kiterjesztését tervezzük automatikus és félautomatikus módszerekkel. Egy tipikus lemma így néz ki:

```
102 átenged V pass concedo przepuścić : LET[DAT HAS ACC]
```

Mint látható, a definíció nem általában az összes *átenged* vagy *pass* írásképű szóra vonatkozik, hanem csupán a magyarban a *vki vkinek vmit* vonzatkerettel egyértelműsített fogalommal. Ha úgy tetszik, az *A vert hadsereg átengedte a várost az ellenfélnek* 'concedo' típusú mondatokat e definícióval előnyben részesítjük az *Az üveg átengedi a fényt* 'transmitto' típusú mondatokkal szemben – hogy mikor melyik jelentést választjuk, és milyen elvek alapján, arról majd a 2. részben lesz szó. Itt tárgyaljuk majd a definíciós (az írott változatban a : után eső) részt is – e bevezetéshez elég annyi, hogy ezek a definíciók egy olyan formális modell elemei, melynek megvan a saját belső szintaxisa és szemantikája.

Mitől fogalmi szótár a 4lang, mit jelent számunkra a 'fogalom'? Ideákról, az emberek fejében megjelenő konceptumokról van szó. Nyelvfilozófiai tekintetben a 4lang gyökerei Platón ideaelméletétől a skolasztikus fogalomfelfogáson át Locke-ig [1] és Fregéig [2] vezethetők vissza. Formális modellekkel dolgozunk, de egyáltalán nem utasítjuk el azt a pszichologizmust, ami a modern generatív elméletben 'kognitív szemantika' néven ismert iskolát jellemzi [3,4,5,6,7,8]. Éppen ellenkezőleg, a formális modelleknek valós, legalábbis felfogásunk szerint valós, tárgya van, t.i. az emberek fejében található ideák.

Közismert, hogy a nyelvtudomány egy fontos szakaszát, a késői strukturalizmustól a generativizmus fellépéséig, áthatotta egyfajta szélsőséges behaviourizmus, mely szerint az emberek fejében vagy nincsenek fogalmak, vagy ha

lennének is, ezek teljességgel megismerhetetlen és szubjektív dolgok melyeknek a tudományban nincs helye. Roy Harris [9,10,11] több kötetet szánt annak az általa *telementáció*-nak nevezett elméletnek a bírálatára, mely szerint az emberek fejében gondolatok vannak, és a nyelv ezeket közvetíti. Úgy véljük, hogy az ilyen és hasonló (neo)behaviourista bírálatokat nem további spekulációval lehet a leghatékonyabban cáfolni, hanem olyan számítógépes modellek építésével, amelyek hagyományosan a nyelvi megértés körébe sorolt tevékenységre képesek. Az ilyen modellek ősképe Leibniz *calculus ratiocinator*a, mai megfelelői pedig az olyan kérdésmegválaszoló algoritmusok, mint az IBM Watson rendszere – aki működés közben lát egy ilyet, annak semmi kétsége nem marad afelől, hogy a kérdezőtől ideák jutnak el, nyelvi úton, a befogadóhoz.

2. Az alapszókincs

Már Leibnizet is erősen foglalkoztatta, hogy a hagyományos szótárakban az egyes fogalmak definíciója gyakran körkörös: az első angol szótár, Cawdrey [12] a *heathen*-t úgy definiálja mint 'gentile', a *gentile*-t pedig mint 'heathen'. Ezt írja (idézi Wierzbicka [5]):

> Tegyük fel, hogy kapsz tőlem egy szép summát azzal, hogy Titiustól veheted át; Titius Caiushoz küld; és Caius Maeviushoz; ha mindig így küldözgetnek az egyik embertől a másikig, soha nem kapsz kézbe semmit.

Egy lehetséges kiút egy olyan alapszókincs megadása, hogy minden más szó már ezek segítségével definiálható. Erre sok próbálkozás volt – a korai elképzelésekről remek áttekintést ad Eco [13]. A modern kísérletek közül a legfontosabb az Ogden [14] által bevezetett Basic English, mely mindössze 850 szóból áll. Ezt használja, legalábbis ezt igyekszik használni [15] a Wikipédia "egyszerű" kiadása (simple.wikipedia.org) is. A nyelvészetben igen jól ismert Swadesh-lista [16] (255 szó) eredetileg nem alapszókincsnek, hanem glottokronológiai vizsgálatokhoz készült, de miután ennél a feladatnál alapvető, hogy a listába vett szavak minden nyelvben előforduljanak, ez a széleskörű elterjedtség már önmagában is biztosítéka annak, hogy a Swadesh-lista szavai az alapszókincsből kerüljenek ki.

Az első modern lexikográfiai elveken alapuló szótár, ami az alapszókincs elvét következetesen végigvitte, a Longman Dictionary of Contemporary English (LDOCE, l. [17]) volt, és a 4lang gerincét is az itt használt Longman Defining Vocabulary (LDV, mintegy 2000 szó és kötött morféma) adja. Ezt egészítettük ki néhány olyan klasszikus listával, mint a Diederich [18] által összeállított 300 leggyakoribb latin szó listája, Whitney [19] szanszkrit gyöklistája, és több magyar, illetve angol gyakorisági vizsgálat leggyakoribb szavai. Természetesen nincs szó arról, hogy az LDOCE definiensei kizárólag az LDV elemeit tartalmazzák, mert a szótár alkotói igen gyakran éltek az indirekt definíciós módszerrel, pl. *Saturn: the* planet *which is 6th in order from the sun and is surrounded by large rings.* De mindaddig, amíg a kiemelt elemek már az LDV által is definiálva vannak, esetünkben *planet: a large body in space that moves round a star, esp. round*

the sun, addig a körkörösség veszélye nem áll fenn, hiszen a második definíciót az elsőbe helyettesítve azt nyerjük: *Saturn: the large body in space that moves round the sun and is the 6th such large body, and is surrounded by large rings* – ez kétségkívül körülményesebb, de ugyanazt jelenti.

Sajnos nincs szó arról, hogy az LDV már önmagában alkalmas lenne fogalmi szótárnak, hiszen ehhez garantálni kell, hogy a szavaknak ugyanabban az értelemben (például *round* nem 'kerek' hanem 'körbe') forduljanak elő a definiendum, mint a definiens oldalon. Garantálni kellett azt is, hogy a szavak minden előforduló kombinációja (pl. *round + up* 'összeterelés, razzia') is definiálásra kerüljön minden olyan esetben, ha definiensben is előfordul. Ez az eset nem is annyira az igekötős igéknél (*phrasal verb*), mint az egyszavas morfológiai összetételeknél (pl. az *-er, -ist* alkotta nomen agentiseknél) fordul elő gyakran. Végső soron a teljes rendszer körmentességét csak a definíciók formális nyelvi eszközökkel való megragadása és gépi elemző építése tette lehetővé.

Külön hangsúlyozzuk, hogy a cél nem a teljes körmentesség, hanem csupán az *uroborosz tulajdonság*, tehát az, hogy definiendán kívüli elem ne legyen egyetlen definiensben sem. Az természetesen elképzelhető, hogy vannak olyan elemek, amelyeket primitíveknek kell tekintenünk (nincs hozzájuk definiens), illetve olyan párok vagy n-esek, melyek kikerülhetetlenek egymás definíciójában: a *férfi*-t nehéz a *nő*-től, a *nő*-t pedig nehéz *férfi*-től függetlenül definiálni. Lássunk néhány összetettebb példát. A *fegyenc* olyan ember, akit fegyőrök fegyházban tartanak, a *fegyőr* pedig olyan, aki fegyenceket tart fegyházban. Mi a *fegyház?* Olyan hely, ahol a fegyőrök fegyenceket tartanak. A három szó egyike helyre, a másik kettő személyre utal, de mind a három csupán ebben a konstrukcióban nyeri el az értelmét. Hasonlóképpen, mi a *tojás,* ha nem az amit a tojó tojt, és mi a tojó, ha nem az, ami tojást tojik? A konceptuális szótárnak nem feladata eldönteni, hogy melyik volt előbb.

Elvben fogalmak bármelyik L listájából kiindulhatnánk, és vizsgálhatnánk hogy ezek definíciójában mely $D(L)$ fogalmak szerepelnek. Az így nyert listát tovább vizsgálva jutunk a $D(D(L)), D(D(D(L))), \ldots$ listákhoz, és azt állítjuk, hogy a folyamat már néhány lépés után konvergál, és a (nyilvánvalóan uroborosz tulajdonságú) fixpont belül marad a 4lang keretein. Tulajdonképpen mindegy is, hogy az egy nyelv szókincsében leggyakoribb szavak jelölte fogalmakból, a legtöbb nyelvben előforduló fogalmakból, a nyelvelsajátítás során legkorábban megjelenő szavakból, a diakrón nyelvfejlődésben legkorábban megjelenő szavakból, vagy akár egy teljesnek szánt szótári listából dolgozunk: definíciós módszereink garantálják, hogy a 4lang körén kívül eső elemre soha nem lesz szükség.

Az alapszókincs tehát a minimális uroborosz tulajdonságú fogalomlista, amely tartalmazhat primitíveket (ilyenek lehetnek pl. a *toj* vagy a *fegy* gyökök), de akár a *függőleges* szó is, melynek definícióját nem újabb nyelvi elemek, hanem a fejünkbe eleve beépített vesztibuláris rendszer adja meg. Hangsúlyozzuk, hogy a primitívségnek nem feltétele a monomorfémikus nyelvi alak, hiszen fogalmi szótárról van szó, a nyelvi alakok csupán adatbáziskulcsként szolgálnak. Ezek közül is kitüntetünk egyet, a *nyomtatási nevet* (*printname*), amellyel az

elemre írásban is és a szoftverben is hivatkozni lehet. Ez lehetne akár az elem sorszáma is, de a programhibák javítását nagyban megkönnyíti, ha mnemonikus értéke van, ezért a kiinduló példánk nyomtatási neve nem *102*, hanem `pass`. A leírásban történeti okok miatt az első helyen a magyar kulcs szerepel, de a rendszer web 2.0 alapú kiterjesztésébe és javításába igencsak nehéz lenne más anyanyelvűeket bevonnunk, ha az azonosítók magyarul lennének. (A cikkben vegyesen hozunk magyar és angol példákat is.)

Már az LDV is túllépett a hagyományos szólistafelfogáson annyiban, hogy nemcsak szavakat (szabad morfémákat), hanem kötött morfémákat is tartalmazott. A 4lang is tartalmaz kötött morfémákat (mind affixumokat, mind gyököket), de a fogalmi rendszer teljességéhez hozzátartoznak azok a szabályok is, melyekkel a morfológiai összetétel során kialakuló jelentést is származtatni tudjuk az összetevők jelentéséből.

3. A fogalmak és a szavak viszonya

A fogalmi szótár központi eleme a definíció. Ezt támogatja, amennyiben ez egyáltalán szükséges, a grammatikai (mind feno-, mind tektogrammatikai, l. [20]) típusra vonatkozó információ. A fenogrammatikai információt a szófajban tartjuk számon, a tektogrammatikai (argumentumstruktúrára vonatkozó) információ pedig a definícióban előforduló mélyesetekből lesz kiolvasható. De a hagyományos értelmező szótáraknak van számos olyan eleme is, amivel a fogalmi szótár eleve nem foglalkozik. A legfontosabb ezek közül a fonológiai információ, amely a ritka hangfestő/hangutánzó esetektől eltekintve a fogalom megértéséhez semmivel nem visz közelebb, de ugyanez vonatkozik általában a morfológiai/morfoszintaktikai információra is. Az angol *go* és *walk* szavak által jelölt, egyébként igen hasonlatos, fogalmak megértéséhez nem visz közelebb az az ismeret, hogy az előbbinek rendhagyó a múlt ideje, de az utóbbinak nem. Nem foglalkozunk a szavak etimológiájával sem, bár ez gyakran segíti a megértést, de úgy véljük, hogy a nyelvelsajátítónak etimológiai információ tipikusan nem áll rendelkezésére, tehát egy ilyeneken alapuló rendszertől semmiféle kognitív realitást nem várhatunk. Nem foglalkozunk a szavak stiláris értékével sem, mert a fogalmi szótár számára elégséges, ha a *kutya* definiálásra kerül, az *eb* már használhatja ugyanezt a definíciót. Megjegyezzük, hogy az általunk figyelmen kívül hagyott szótári információk az átlagos szócikk kevesebb, mint 10%-át teszik ki, akár bitben, akár nyomtatott karakterben számolva.

A definíció célja a fogalmak közti belső kapcsolatok rögzítése. Amikor a kapcsolat csupán történeti, mint pl. *bishop* ‘püspök’, illetve *bishop* ‘futó (sakkfigura)’ akkor az értelmező szótárakban szokásos módon alsó indexekkel különböztetjük meg a lemmákat. Az ilyesfajta tiszta homonímiák elkülönítése a poliszémiától már a hagyományos lexikográfiában is sok fejtörést okoz, és a nehézségek alól természetesen a fogalmi szótár sem tudja teljesen kivonni magát. Szerencsére a fogalmi szótárnál érvényesíteni tudunk több olyan tényezőt, amely a feladatot lényegesen megkönnyíti. Az első a rugalmasabb szófajkezelés: ezt lentebb a *fagy* ige, ill. *fagy* főnév példáján fogjuk illusztrálni.

A második tényező módszertani. Kirsner [21] két élesen szemben álló megközelítésről ír: a *poliszemikus* felfogás igyekszik a szavak jelentéseit maximálisan elkülöníteni, pl. *bachelor*$_1$ 'nőtlen felnőtt férfi', *bachelor*$_2$ 'pár nélküli fóka', *bachelor*$_3$ 'más lovag zászlaja alatt szolgáló lovag', és *bachelor*$_4$ 'BA vagy BSc fokozattal rendelkező személy'. A *monoszemikus* megközelítés (melyet Kirsner *Saussure*-i és *Columbia School* megközelítésnek is nevez) egy általános, absztrakt jelentést tételez, mely alá esetünkben legalább az első három aljelentés besorolható: 'tipikus férfiszerepben kielégítetlen'. A 4lang a monoszemikus megközelítést követi, ennek filozófiai alapjairól l. [22].

A poliszémia minimalizálásának van még egy igen fontos módszertani indoka, amely véleményünk szerint nemcsak a 4lang, hanem minden fogalmi szótár lényegéből ered. Egy ilyen szótár célja kifejezetten a szavak egymás közti viszonyainak, nem pedig a szavak és a világ dolgai közti viszonyok feltárása. Élesen el kell különíteni a *lexikai* és az *enciklopédikus* információt, alapjában ugyanazon kritériumok mentén, amelyekkel a filozófia az analitikus és a szintetikus kijelentéseket választja szét. Az értelmező szótári gyakorlatban az enciklopédikus információ gyakran keveredik a lexikaival: példaképp álljon it a *potash* 'hamuzsír, kálisó' definíciója a Webster's Third-böl:

> 1a: potassium carbonate, esp. that obtained in colored impure form by leaching wood ashes, evaporating the lye usu. in an iron pot, and calcinating the residue – compare pearl ash. b: potassium hydroxide. 2a : potassium oxide K_2O in combined form as determined by analysis (as of fertilizers) ⟨ soluble ∼ ⟩ b: potassium – not used systematically ⟨ ∼ salts ⟩ ⟨ sulfate of ∼ ⟩ 3: any of several potassium salts (as potassium chloride or potassium sulfate) often occurring naturally and used esp. in agriculture and industry ⟨ ∼ deposits ⟩ ⟨ ∼ fertilizers ⟩

Jól látható, ahogy az enciklopédikus tudás rögzítésével a szótáríró magának csinálja a poliszémiát. Az LDOCE felfogása szerint itt egyáltalán nincs szó többértelműségről:

> any of various salts of potassium, used esp. in farming to feed the soil, and in making soap, strong glass, and various chemical compounds

és a 4lang, ha a hamuzsírt az alapszókincs részének tekintené, akkor még tovább menne az absztrakcióban:

```
salt, HAS potassium
```

A modern tudásreprezentációs sémák erősen hajlanak a tudományközpontúság felé, pl. Kripke [23] a vizet mint H_2O-t definiálja. Történetileg azonban a nyelvi tények megelőzik a tudományos ismereteket, előbb volt a hamuzsír, mint a kálium. Nincs semmi okunk azt feltételezni, hogy egy minden tekintetben kielégítő biológiai definíció hiányában az emberek nem tudják mondjuk a *kutya* szót használni, vagy hogy Berzelius előtt a *víz* mást jelentett, mint ma. Mivel a fogalmi szótár célja nem az egyes fogalmak meghatározása, hanem ezek rendszerének feltárása, a *víz* definíciója nem H_2O, hanem

`2622 víz N water aqua woda: liquid, NOTHAS colour, NOTHAS taste, NOTHAS smell, life NEED`

tehát csupa olyasmi, amit az emberek évezredek óta tudnak (és amiknek a modern tudomány akár ellent is mondhat). Ahol mégis a tudományok által definiált dolgokról van szó (ilyen pl. a *kálium*, amelynek egyszerűen nincs hétköznapi definíciója) ott egy külső enciklopédiára, konkrétan a Wikipédiára mutató kereszthivatkozásokat használunk:

`potassium : element, @http://en.wikipedia.org/wiki/Potassium`

Ugyanilyen kereszthivatkozásokat használunk ott is, ahol a lexikográfiai gyakorlat illusztrációkkal dolgozik – ez is meglehetősen ritka eset, pl. az angolszász lexikográfia alapműve, az Oxford English Dictionary egyáltalán nem használ illusztrációkat, a Websters Third pedig a szócikkek kevesebb mint fél százalékánál.

A definíciók szintaxisa arra a feltevésre épít [24], hogy a primitívek listája egyáltalán nem kell, hogy kettőnél több argumentumú (ditranzitív vagy magasabb aritású) elemeket tartalmazzon, mert ezeket mindig lehet egyszerűbb aritásúakkal definiálni. Jó példa a `give`, aminek a definíciója 'cause to have', egész pontosan `CAUSE[DAT HAS ACC]` – a rendszerbe beépített redundanciaszabály szerint a `CAUSE` alanya, mint minden tranzitív predikátumé, nominatívuszi. Az alapfogalmak túlnyomó része intranzitív, ilyenek a köz- és tulajdonnevek (kivéve természetesen a relációs főneveket), a melléknevek, és az intranzitív igék is. A tranzitív elemeket az írott változatban csupa nagybetűvel jelöljük. Az implementáció alapját adó gépek (machine, definícióját l. [25]) kétféle változatát használjuk: egy-, illetve kétpartíciósat (erről bővebben l. [26]), attól függően, hogy az elemet intranzitívnak vagy tranzitívnak tekintjük.

Az intranzitív elemeket mint a rájuk analitikusan jellemző predikátumok konjunkcióját definiáljuk, pl. `488 düh N anger furor gniew: feeling, bad, strong, aggressive`. Annyiban Arisztotelészt és a skolasztikus hagyományt követjük, hogy a definíciónak az esszenciát kell megragadnia, de abban eltérünk a hagyománytól, hogy mi a düh *szó*, a düh *fogalom* jelentését, nem pedig a világban található reális düh lényegét próbáljuk megragadni. Ez utóbbi nyilván valami hormonszintváltozással függ össze, de ezt mi enciklopédikus ténynek tekintjük, és mint ilyet figyelmen kívül is hagyjuk. Ebből adódik a 4lang egy fontos tulajdonsága: számunkra a *dobermann* és a *pincsi* definíciója egyaránt `dog`.

Kevesebb, mint harminc primitív tranzitív elemünk van, ezek között a legfontosabbak grammatikai jellegűek. A legnagyobb csoport a mélyesetek `NOM, ACC, DAT, ...`, de a melléknevek fokozásánál elkerülhetetlen az `ER`, és a főnevek birtoklásánál kikerülhetetlen egy `HAS` alak. A tisztán konceptuális binárisok közt a leggyakrabban az `AT` szerepel definiensben, ebben a monoszemikus felfogásnak megfelelően együtt szerepel az időbeli és a térbeli összekapcsolódás. A tranzitívnál bonyolultabb argumentumstruktúra megragadásának eszköze a tranzitív relációk egymásba ágyazása, pl. `1846 öl V kill interficio zabijać: CAUSE[ACC[die]]`. Ebben a tekintetben a 4lang a generatív szemantika definíciós módszereit követi, a különbség elsősorban a változók és a változókötés mechanizmusának sajátos, gépeken alapuló megvalósításában áll.

Térjünk most vissza az olyan többszófajú elemek problémájára, mint az angol *divorce* vagy a magyar *fagy*. Felfogásunk szerint ilyenkor az igénél és a főnévnél ugyanarról a fogalomról van szó, t.i. arról a folyamatról, amiben a víz szilárd lesz, vagy ennek okáról: definíciós nyelvünkön `cold CAUSE, before[liquid], after[solid,<ice>]`. A természetes nyelv egy sajátos jellemzője, hogy az okot és az okozatot ilyenkor nem szemantikai, hanem fenogrammatikai eszközökkel különíti el. A tökéletes filozófiai nyelv kialakítására törekvő filozófusokat, pl. Francis Bacont, ez és a többi *idola fori* nagyon zavarta, de véleményünk szerint a szemantika a nyelvtudomány része, és mint ilyen deskriptív, nem pedig normatív módszertannal dolgozik.

A 4lang meghoz számos olyan technológiai döntést, amelyeket minden fogalmi szótárnak meg kell hoznia, de nem feltétlenül úgy, ahogy ezt mi tesszük. Ilyen az alapértelmezett (*default*) értékek konzekvens használata: az előző példánál maradva a *fagy* eredménye alapértelmezésben a jég, bár természetesen nagy hidegben az alkohol, a paraffin, de még a hőmérő higanya is megfagy. A szótárban a default értékeket `< >` jelöli. Egyedi döntés az is, hogy a *before* és *after* elemek egyváltozósak, hiszen másik változójukat úgyis a cselekvés idejéhez kellene kötnünk. Végül ugyanilyen döntés az is, hogy kikerültük az uniform Boole-jellegű negációt, helyette külön primitívnek véve a `NOTHAS, NOTAT` és hasonló negatív relációkat: pl. a *kígyó* definiáló tulajdonsága a `NOTHAS leg`, a *lélek*-nek a `NOTHAS material`, a *lop*-nak pedig a `NOTHAS right`. Van természetesen negációs primitív (intranzitív) elem, sőt többféle is van, ezek közül legfontosabb a `lack` amely normálisan (alapértelmezésben) meglévő elem hiányát jelzi: például a *beteg* `lack(health)`, ami több, mint a `NOTHAS health`, hiszen nem csak arról van szó, hogy nincs neki, hanem egyben arról is, hogy kellene lennie, míg ez utóbbi következtetést pl. a kígyó lábáról nem kívánjuk levonni.

4. Alkalmazások

A 4lang adja az alapját több olyan rendszernek is, melyeket munkacsoportunk már a gyakoratban is bemutatott: ilyen a SHRDLU 2.0 (Kutatók Éjszakája 2011), az Elvira-asszisztens (Edinburgh 2012), és a robotpénztáros (Kutatók Éjszakája 2012). Mint minden gyakorlatban működő rendszernél, itt is szükség van interfészekre, amelyek a rendszeren kívüli komponensek (a kockarakosgató robot, a www.elvira.hu weblap, illetve a pénztári adatbázis) meghajtására alkalmasak. A rendszer egésze tehát képes az ilyen és hasonló külső komponensekkel kapcsolatos enciklopédikus tudás megragadására is, de ezt formailag is eltérő, nem gépeken, hanem attribútum-érték mátrixokon (AVM) alapuló mechanizmussal teszi.

Tekintsük például az Elvira-asszisztenst, amely a www.elvira.hu-ról azt tudja, hogy ha három attribútum (dátum, kiindulás, cél) közül legalább a kiindulás és a cél már ki vannak töltve, akkor az ezekből kialakított queryt `http put` segítségével elküldi a www.elvira.hu-ra. A természetes nyelvi rendszer feladata kettős: egy magyarul megfogalmazott kérdésből, pl. *Mikor megy holnapután vonat Szegedre a Nyugatiból?* észre kell vennie, hogy ez egy olyan kérdés, amit

az Elvira meg tud válaszolni, másrészt hogy ki tudja választani az egyes attribútumokra vonatkozó értékeket: dátum: holnapután, kiindulás: Budapest Nyugati, cél: Szeged.

Ehhez a 4lang egy olyan változatára van szükség, ami a *holnap* mellett (ez benne van az alapszókincsben) tartalmazza a *holnapután* szót is, és persze a *vonat* szót is. Felhasználásra kerül a szótárnak néhány olyan eleme is, amely a példamondatban ugyan nem szerepel. de kikerülhetetlen közbenső kapocs az Elvirához: ilyen elsősorban az *Elvira* szó, ami definíciója szerint `vonat, menetrend` és enciklopédikus részében tartalmazza a fentebb leírt háromelemű AVM-et. Nyilvánvaló, hogy a rendszer csak akkor tudja hívni az Elvirát, ha tudja, hogy van ilyen. A felhasználónak viszont nem kell ezt tudnia, kiinduló mondatunk nem az, hogy *Kérdezd meg az Elvirát...*

Rendszerünk logikájából adódóan szükség van még a *vonat* és a *menetrend* szavak definíciójára is, de ezekben már semmi Elvira-specifikus nincsen: a vonat számunkra `mass_transit, rail, ...` a menetrend pedig egyszerűen `mass_transit, when`. Az Elvira AVM-hez egy teljesen általános mechanizmussal, a terjedő aktivációval (*spreading activation*, l. [27]) jutunk el az eredeti inputban szereplő *mikor* (when), illetve *vonat* (train) szavakon, illetve az inputban már nem szereplő, de ezek által aktivált *menetrend* (schedule) szón keresztül.

Köszönetnyilvánítás

A 4lang-ot használó rendszerek kialakításán legtöbbet Nemeskey Dávid, Recski Gábor, és Zséder Attila (SZTAKI) dolgoztak. A 4lang alapjait, illetve az egyes definíciókat illetően számos hasznos tanácsot kaptunk még az alábbiaktól: Kálmán László (NYTI), Muntág Márton (ELTE), Rebrus Péter (NYTI), Rung András (KREA), Szakadát István (BME MOKK), Szőts Miklós (ALL), Varasdi Károly (PPKE), Vásárhelyi Dániel (ELTE). A munka az OTKA Szemantikai Alapú Nyelvtechnológia (82333) pályázatának támogatásával készült.

Hivatkozások

1. Locke, J.: An Essay Concerning Human Understanding. Ward, Locke and Bowden (1689)
2. Frege, G.: Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. L. Nebert, Halle (1879)
3. Jackendoff, R.S.: Semantic Interpretation in Generative Grammar. MIT Press (1972)
4. Lakoff, G., Johnson, M.: Metaphors we live by. University of Chicago Press (1980)
5. Wierzbicka, A.: Lexicography and conceptual analysis. Karoma, Ann Arbor (1985)
6. Talmy, L.: Force dynamics in language and cognition. Cognitive science **12**(1) (1988) 49–100
7. Langacker, R.: Foundations of Cognitive Grammar. Volume 1. Stanford University Press (1987)
8. Langacker, R.: Foundations of Cognitive Grammar. Volume 2. Stanford University Press (1991)

9. Harris, R.: The language-makers. Duckworth (1980)
10. Harris, R.: The language myth. Duckworth (1981)
11. Harris, R.: The language machine. Duckworth (1987)
12. Cawdrey, R.: A table alphabetical of hard usual English words. (1604)
13. Eco, U.: A tökéletes nyelv keresése. Atlantisz (1998)
14. Ogden, C.: Basic English: a general introduction with rules and grammar. K. Paul, Trench, Trubner (1944)
15. Yasseri, T., Kornai, A., Kertész, J.: A practical approach to language complexity: a Wikipedia case study. PLoS ONE (2012)
16. Swadesh, M.: Salish internal relationships. International Journal of American Linguistics **16** (1950) 157–161
17. Boguraev, B.K., Briscoe, E.J.: Computational Lexicography for Natural Language Processing. Longman (1989)
18. Diederich, P.B.: The Frequency of Latin Words and Their Endings. Illions, The University of Chicago Press (1939)
19. Whitney, W.: The Roots, Verb-forms, and Primary Derivatives of the Sanskrit Language. Motilal Banarsidass (1845)
20. Curry, H.B.: Some logical aspects of grammatical structure. In Jakobson, R., ed.: Structure of Language and its Mathematical Aspects. American Mathematical Society, Providence, RI (1961) 56–68
21. Kirsner, R.: From meaning to message in two theories: Cognitive and saussurean views of the modern dutch demonstratives. Conceptualizations and mental processing in language (1993) 80–114
22. Ruhl, C.: On monosemy: a study in lingusitic semantics. State University of New York Press (1989)
23. Kripke, S.A.: Naming and necessity. In Davidson, D., ed.: Semantics of Natural Language. D. Reidel, Dordrecht (1972) 253–355
24. Kornai, A.: Eliminating ditransitives. In: Formal Grammar. (2011) 243–261
25. Eilenberg, S.: Automata, Languages, and Machines. Volume A. Academic Press (1974)
26. Kornai, A.: The algebra of lexical semantics. In Ebert, C., Jäger, G., Michaelis, J., eds.: Proceedings of the 11th Mathematics of Language Workshop. LNCS 6149. Springer (2010) 174–199
27. Quillian, M.R.: Semantic memory. In Minsky, ed.: Semantic information processing. MIT Press, Cambridge (1967) 227–270

Hunglish mondattan – átrendezésalapú angol–magyar statisztikai gépifordító-rendszer

Laki László János[1,2], Novák Attila[1,2], Siklósi Borbála[2]

[1] MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
[2] Pázmány Péter Katolikus Egyetem,
Információs Technológiai Kar,
Budapest, Práter u. 50/a,
e-mail:{laki.laszlo,siklosi.borbala,novak.attila}@itk.ppke.hu

Kivonat A napjainkban népszerű frázisalapú statisztikai gépifordító-rendszerek az egymáshoz hasonló szerkezetű és a nem nagyon gazdag ragozó morfológiával bíró nyelvpárok esetében látványos eredményeket értek el az utóbbi évek során. Azon nyelvpárok esetében azonban, ahol jelentős szórendi és strukturális különbségek vannak a két nyelv között, az eredmények messze elmaradnak a várakozásoktól. Az utóbbi kategóriába tartozik az angol-magyar nyelvpár is. Cikkünkben egy olyan angol-magyar statisztikai gépifordító-rendszer létrehozására tett kísérletünket írjuk le, amelyben a két nyelv közötti strukturális különbségeket úgy próbáltuk áthidalni, hogy az angol forrásnyelvi mondatok szintaktikai elemzését felhasználva, azokat automatikusan a nekik megfelelő magyar mondatok szerkezetének jobban megfelelő szórendűvé alakítottuk. A korlátozott mértékű tanítóanyag és a magyar ragozó jellege miatt fennálló adathiány-probléma megoldása érdekében szó- helyett morfémaalapú fordítórendszert hoztunk létre.

Kulcsszavak: SMT, morfológiai elemzés, átrendezés

1. Bevezetés

Az informatika fejlődése új lehetőségeket nyitott meg többek közt a nyelvészetben. A humán nyelvtechnológia egyik legfontosabb feladata, hogy leküzdje a soknyelvűség okozta akadályokat és nehézségeket, illetve támogassa globalizálódó világunk különböző nyelveinek megértését. Ennek megvalósításában nyújt nagy segítséget a gépi fordítás.

Az első ilyen rendszerek előre definiált szabályok, illetve transzformációk alapján működtek. A szabályalapú gépi fordítás hátránya, hogy a különböző nyelvi sajátosságok nem írhatók le mindent lefedő szabályrendszerrel. A statisztikai módszeren alapuló gépi fordítás (SMT) a számítógépre bízza a szabályrendszer felépítését, ami egy párhuzamos kétnyelvű korpusz felhasználásával történik.

Azokra a nyelvekre, melyek szintaktikailag hasonlóak és morfológiailag nem túl komplexek, a frázisalapú SMT módszerei viszonylag jó eredménnyel működnek. Ezzel ellentétben az ilyen szempontból egymástól távolabb eső nyelvpárok

(pl. angol-magyar) esetén jelentős lemaradás van. Több tanulmány bemutatta azt is, hogy az ilyen esetekben csupán a tanító korpusz növelése nem elegendő a minőség számottevő javításához. A magyar nyelv szabad szórendje és szóalaki sokfélesége miatt nem is lehetséges olyan korpusz létrehozása, amely minden nyelvi jelenséget elég jól lefedne. Ezért célunk egy olyan hibrid fordítórendszer létrehozása volt, amely amellett, hogy kihasználja a statisztikai gépi fordítás előnyeit, igyekszik csökkenteni a szórendi különbségekből és a magyar nyelv morfológiai sokszínűségéből adódó problémákat.

2. Gépi fordítás angol-magyar nyelvpárra

Bár a gépi fordítás területén vannak jelentős eredmények, de a feladat korántsem tekinthető megoldottnak. Különösen igaz ez az egymástól távol álló nyelvpárok esetén, mint például a magyar és az angol. A legfőbb probléma a két nyelv jelentősen eltérő struktúrájában rejlik, emberként is leginkább csak az egyik nyelven megfogalmazott mondat jelentésével azonos jelentésű mondatot tudunk megfogalmazni a másik nyelven, ez azonban nem nyelvtani megfeleltetést jelent. Ez a probléma az alkalmazott fordítási módszer kiválasztásánál éppen úgy jelen van, mint a fordítás folyamán, vagy a kiértékelésnél. A tapasztalatok alapján a gépi fordítás minőségét három tulajdonság befolyásolja a forrás- és célnyelvek megválasztásának függvényében: a szórendbeli eltérés mértéke, a célnyelv nyelvtani összetettsége és a két nyelv közötti történeti kapcsolat [1].

2.1. A magyar nyelv sajátosságai

A magyar és az angol nyelvek mind történetileg, mind nyelvtanilag egymástól távol álló nyelvek, illetve célnyelvként a magyart tekintve, ennek összetett volta sem kérdéses. A legfontosabb nehézséget a magyar nyelvre jellemző gazdag morfológia, és a két nyelv jelentősen eltérő szórendje jelenti, amelyet a magyarban egész más tényezők határoznak meg, mint az angolban.

A magyar nyelvre jellemző agglutináló (ragozó) jelleg toldalékok halmozását is lehetővé teszi. Szintén jellemző a többféle alakváltozat mind a szótövek, mind a toldalékok terén, a gazdag esetrendszer és az irányhármasság (honnan? hol? hová?) a helyhatározók használatában. Kevés az igeidő, és hiányzik az indoeurópai nyelvekre jellemző birtoklást kifejező ige („én birtoklok valamit" helyett „nekem van valamim"). A magyar nyelv megkülönbözteti a határozatlan („alanyi") és a határozott („tárgyas") ragozást: olvasok, olvasom; a főnévi igenév pedig ragozható (látnom, látnod, látnia stb.) . A magyar az indoeurópai nyelveknél sokkal ritkábban használja a határozatlan névelőt (egy); a páros szerveket (pl. kéz, láb, szem, fül) és a több birtokos egy-egy birtokát is egyes számban mondja (pl. élik az életüket, nem pedig életeiket) a számneves névszói csoportok pedig alakilag egyes számúak, és így is egyeztetjük őket az igével.

Egy egyszerű magyar mondatban általában az alany az első, az ige a második és végül a tárgy az utolsó elem. A magyar pro-drop nyelv, az alanyi és tárgy névmások is rendszerint hiányoznak a mondatból . Mindemellett, a szórend nem

feltétlenül kötött, ugyanis azt elsősorban nem szintaktikai szabályok, hanem pragmatikai tényezők határozzák meg (ugyanakkor semleges mondatok esetében rendkívül összetett szintaktikai megszorítások is). Így például gyakran előfordul a tárgy-alany-állítmány vagy az állítmány-alany-tárgy sorrend is, hiszen a ragozás egyértelműen utal az elemek mondatbeli szerepére. A kihangsúlyozni kívánt információ rögtön a ragozott ige elé helyezendő.

A magyarral ellentétben az angol főleg izoláló nyelv, vagyis a mondatokban a fő nyelvtani funkciókat a szavak sorrendje határozza meg. Szórendje sokkal kötöttebb, mint a magyar nyelv esetén, jellemzően alany-állítmány-tárgy alakú. Ugyanakkor számos példáját hordozza a flexiónak (a tő megváltoztatásával járó ragozás), főleg a rendhagyó esetekben. Így bár a nyelv flektálónak tekinthető, lassan tart az izoláló felé, hiszen például a mai angolban már nincsenek esetek, az eredeti esetragok lekoptak [2].

2.2. Statisztikai gépi fordítás

Napjaink nemcsak legelterjedtebb módszere a statisztikai gépi fordítás (SMT), hanem egyben a legtöbb lehetőséget magában rejtő és az egyik legjobban kutatott irányzat is. Bár a statisztikai módszereket nyelvfüggetlen megoldásoknak tekinthetjük, azonban mégis szükségesnek látjuk a nyelvspecifikus problémák kezelését, melyet elő- és utófeldolgozási lépésekként építettünk be a rendszerbe.

A statisztikai gépi fordítás alapötlete a kétnyelvű párhuzamos korpuszból tanult statisztika alapján való fordítás, illetve az így nyert fordítási lehetőségek célnyelvi korpuszból épített, a célnyelvet jellemző modell alapján történő kiértékelése.

A fordítás során a mondat, amelyet le szeretnénk fordítani (forrásnyelvi mondat) az egyetlen, amit biztosan ismerünk. Ezért a fordítást úgy végezzük, mintha a célnyelvi mondatok halmazát egy zajos csatornán átengednénk, és a csatorna kimenetén összehasonlítanánk a forrásnyelvi mondattal.

Ezt a folyamatot a Bayes-tétel segítségével lehet leírni két valószínűségi változó szorzataként. Ezeket fordítási- és nyelvmodellnek nevezzük. Az a mondat lesz a rendszerünk kimenete, amelyik a legjobban hasonlít a fordítandó (forrásnyelvi) mondathoz.

3. Átrendezési szabályok alkalmazása előfeldolgozási lépésként

A fent részletezett nyelvi különbségek áthidalása végett a cikkben bemutatott rendszerben olyan előfeldolgozási lépéseket alkalmaztunk, melyeknek célja a forrásnyelvi (angol) szöveg mondatainak a célnyelvi (magyar) mondatokhoz hasonló alakra hozása. Ehhez első lépésként az angol mondatokra szófaji egyértelműsítés és szintaktikai elemzés után a mondatban megjelenő függőségi relációkat is meghatároztuk. Így olyan gazdag információkkal kiegészített mondatokat kaptunk,

melyek birtokában megfogalmazhatók olyan szabályok, amelyek a mondatok magyar megfelelőjében szereplő szerkezetekkel párhuzamos formára hozzák azokat. Így a fordítórendszer tanítása során a nyelvpárt az alaprendszerénél jobban reprezentáló statisztikák jönnek létre. Mivel a statisztikai módszer alapját a kétnyelvű párhuzamos mondatokban szereplő szavak megfeleltetésére épített valószínűségek képezik, ezért a szóösszerendelés minősége alapjaiban meghatározza a végső fordítás minőségét is.

A két nyelv közelítése a morfémák szavakba szerveződése szempontjából hatékonyan csökkentheti a szóösszerendelési hibák számát. Más nyelvekkel (pl. az angol-német nyelvpárral) kapcsolatban publikált eredmények pedig azt mutatják, hogy a szórend szabályalapú megváltoztatása csökkenti a dekódolás során a fordításból kimaradt szavak számát.

Az alkalmazott szabályaink csak azokat a szórendi eltéréseket szüntetik meg, amelyek a két nyelv között szabályszerűen fellépnek (pl. elöljárók vs. esetragok / névutók), nem volt célunk ugyanakkor a magyar „szabad szórend"-ből adódó eltérések eltüntetése.

A szabályok az angol mondatok szófajilag egyértelműsített elemzését, közvetlen összetevős és függőségi elemzését használják. A függőségi relációkból az elemzés után kiválasztjuk a releváns kapcsolatokat, amelyek mentén alkalmazzuk a megfelelő szabályt. Nagyon egyszerű példa az angol „in my house" kifejezés, mely az átrendezés és összevonások után „house_my_in" formára alakult, amely megfelel a magyar „házamban" alaknak. Az ilyen rövid szókapcsolatok során a szabályok alkalmazása nem jelent nagy problémát, azonban hosszabb mondatok esetén az egymáshoz kapcsolódó részek egészen távol is eshetnek, több függőségi kapcsolatban is érintettek lehetnek. Hasonló módon kerültek beszúrásra olyan morfológiai elemek, melyek az eredeti angol mondatban nincsenek explicit módon jelölve (pl. tárgyrag), a magyar megfeleltetés miatt azonban szükségesek. Természetesen figyelembe vettük az összetartozó szerkezeti egységeket, ezeket az átrendezés során is egységként kezelve, egyben helyeztük át.

Három fő csoportba sorolható átrendezési szabályokat alkalmaztunk:

3.1. Szórendet és morféma összevonást/felbontást tartalmazó szabályok

Ezek a szabályok a függőségi relációk meghatározása után, a közvetlen összetevős szerkezetet is figyelembe véve alakítják át a szavak sorrendjét, ezzel egyidőben vonják is össze azokat, amikor szükséges. Olyan szabályok kerülnek végrehajtásra, mint a passzív, a segédigés, a prepozíciós és birtokos szerkezetek átalakítása, az angolban hátravetett módosítók előremozgatása, és még néhány, ritkábban előforduló szabály. Fontos az átrendezési szabályok végrehajtásának sorrendje is, mivel nem csak szavakat, hanem nagyobb egységeket helyezünk át. Az alábbi mondatban két szabályt hajtunk végre:

A „living in the city” prepozíciós szerkezet a PARTMOD[1] (merchant, living), PREP[1](living, in) és a POBJ[1](in, city) relációk mentén kerül átalakításra. Először a prepozíció kerül rá annak gyerekére, majd az így kapott összevont szót helyezzük át az ezt megelőző főnévi szerkezet elé. Így kialakul az „a város_ban élő” magyar fordításnak már egyértelműen megfeleltethető szórend. Hasonlóan járunk el a „the sons of the merchants” esetén a megfelelő relációk használatával, melynek eredményeként a „kereskedők fiai” magyar szintaktika szerinti alakra jutunk. Ezt látható az 1. táblázatban.

1. táblázat. Példamondat I.

Eredeti mondat:	The/DT sons/NNS of/IN the/DT many/JJ merchants/NNS living/VBG in/IN the/DT city/NN ./.
Átrendezett mondat:	the/DT city/NN_in/IN living/VBG many/JJ merchants/NNS sons/NNS_of/IN ./.

Bár általában az angol oldalon szükséges a szavak számának a csökkentése azok összevonásával, így a magyarnak megfelelő toldalékok létrehozásával, mégis vannak esetek, amikor új szavakat kell beszúrni az átrendezések során az angol mondatba. Mivel az ilyen eseteknél nem tudjuk előre meghatározni az oda illő magyar szót, mivel az az aktuális szövegkörnyezettől függ, ezért csupán egy karaktersorozat kerül beillesztésre, melynek konkrét realizációját a fordítás kell hogy meghatározza. A 2. táblázatban az xxx/xxx jelöli a „lévő” magyar szó pozícióját a mondatban, valamint néhány további átrendezési példát is tartalmaz.

2. táblázat. Példamondat II.

Eredeti mondat:	That/DT is/VBZ the/DT account/NN at/IN the/DT largest/JJS bank/NN in/IN Bern/NNP ./. ”/”
Átrendezett mondat:	That/DT is/VBZ the/DT Bern/NNP_in/IN xxx/xxx largest/JJS bank/NN_at/IN xxx/xxx account/NN ./. ”/”
Eredeti mondat:	Only/RB I/PRP ’m/VBP allowed/VBN to/TO ./.
Átrendezett mondat:	Only/RB allowed/VBN_P_they/P3 I/PRP_acc/ACC to/TO ./.

3.2. Átrendezést nem tartalmazó, csupán a morfológiai összetételt változtató szabályok

Az angol mondatokban sok olyan információ nincs jelen, amely a magyar oldalon toldalékokként szerepelnek. Ezekre azonban a függőségi relációk alapján tudunk

[1] A függőségek teljes listája itt olvasható: http://nlp.stanford.edu/software/dependencies_manual.pdf

következtetni. Így például az angolban jelöletlen tárgyrag a megfelelő relációk mentén meghatározható. Az ilyen esetekben beszúrtuk ezeket a morfémákat az angol mondatba.

Így lett a *„while/IN giving/VBG a/DT present/NN ./."* mondatból *„while/IN giving/VBG a/DT* ***present/NN_acc/ACC*** *./."*

Előfordulnak továbbá olyan esetek is, amikor az angol különálló szóként jelöli a magyar toldaléknak megfelelő morfémákat, amelyeket így rácsatoltunk a megfelelő szóra. Ezek az összevonások nem nagyobb szerkezetek átrendezését jelentik. Például a birtokos névmás esetén, ha a birtok tárgya is szerepel a mondatban, akkor azt csak ennek megfelelően hozzákapcsoljuk ahhoz. Így lett a *„my/PRP$ own/JJ country/NN"* mondatból *„own/JJ country/NN_my/PRP$"*.

3.3. Redundanciák feloldása, utófeldolgozás

Ezek a szabályok elsősorban az első két csoportba tartozó átrendezések mellékhatásai miatt szükségesek. Például előfordulhat, hogy az átrendezés után két névelő kerül egymás mellé, ilyen esetekben az egyiket törölni kell. Ide tartozik még a birtokos *'s* rácsatolása a megfelelő szóra. Ezen kívül még néhány apró módosítást láttunk szükségszerűnek (például pénznemek áthelyezése a számérték utánra).

3. táblázat. Példamondat III.

Eredeti mondat:	John's cat
Függőségi relációk:	poss(cat, John)
	possessive(John, 's)
Átrendezett mondat:	John/NNP cat/NN_'s/POS

4. Felhasznált eszközök

4.1. Korpusz

Az elérhető angol-magyar párhuzamos korpuszok többsége nem alkalmas egy általános SMT-rendszer betanítására, mivel csupán egy-két terület terminológiáját tartalmazzák. Munkánk során ezért az elérhető legnagyobb és témáját tekintve legáltalánosabb párhuzamos korpuszt, a BME MOKK és az MTA Nyelvtudományi Intézete készített Hunglish korpuszt [3] használjuk. Ez a korpusz egyidejűleg több területről tartalmaz szövegeket: szépirodalom, magazin, jog, filmfeliratok. A rendszer betanítása során nehézséget jelent azonban, hogy az egyes részek minősége meglehetősen változó. Az így létrejött korpusz mérete 1202205 párhuzamos mondatpár.

4.2. Szintaktikai és függőségi elemző

Az előfeldolgozás első lépéséhez szükség volt egy robusztus szófaji egyértelműsítőre, szintaktikai és függőségi elemzőre az angol mondatok átrendezéséhez, illetve a morfémaalapú fordítás miatt a magyar nyelv elemzésére.

Magyar nyelvre a PurePos[4] automatikus morfológiai annotáló eszközt használtuk. A teljes tanító anyag magyar oldalát ezzel elemeztük, az összetett morfológiával rendelkező szavak esetén ezeket felbontottuk elemi egységekre azért, hogy az angol oldalon külön szóként, a magyar oldalon azonban csak toldalékként megjelenő morfémák is megfeleltethetőek legyenek egymásnak. Mivel a morfológia önmagában tartalmaz minden információt a szóalakok eredeti voltáról, ezért a szavak jelentésének megfeleltetésére elegendő azok szótövét figyelembe venni, így mivel az egyes szavakhoz tartozó szótő alakok előfordulási gyakorisága jóval nagyobb a korpuszban, ezért biztosabb statisztikát kaptunk, mint a teljes szóalakokra való statisztika építése során. Természetesen az így betanított morfémaalapú fordítórendszer fordítási eredménye is szótő+címkék alakú eredményt hoz létre, ezért ezeket a fordítás után vissza kell alakítani, amihez a Humor [5] szóalakgeneráló modulját használtuk.

Angol nyelvre a Stanford elemzőt [6] használtuk, mely az egyik gyakran használt szabadon hozzáférhető angol szintaktikai elemző. Az elemző hozzáférhető változatát a Penn Wall Street Journal Treebank egy töredékén tanították. Az elemzés minősége sokkal fontosabb számunkra, mint a gyorsasága, hiszen az elemzés és a szóösszekötés offline, csak a tanítás során egyszer történik (illetve a fordítandó szöveget kell még elemeznünk), ezért úgy döntöttünk, hogy az elemző lexikalizált változatát alkalmazunk. Ez valamivel jobb elemzést eredményezett, mint az alapváltozat, de még így is nagyon sok olyan eset fordult elő, melyeket az elemző nem tudott megfelelően kezelni.

A Stanford parser sorba kapcsolt elemekből álló rendszer, amelynek első szófaji egyértelműsítő komponense önmagában meglehetősen sok hibát generál, amelyet azután minden további komponens csak továbbiakkal tetéz. A korábbi komponensek hibáit a láncban később következők soha nem javítják, inkább a legnyakatekertebb megoldásokkal próbálnak a kapott inputhoz alkalmazkodni. Ezek a rosszul elemzett és rossz helyre csatolt szavak és kifejezések az egész rendszerben kritikus problémát jelentenek, hiszen az átrendezések ez alapján az elemzés alapján történnek. Ez azt jelenti, hogy ha egy eleve rosszul elemzett szöveget rendezünk át, akkor az így kapott hibás átrendezés inkább ront, mint javít a fordítás minőségén.

Az első ilyen hibaforrás a helytelen POS-címkék használata az elemző által ismeretlen szavak, vagy az ismeretlen kontextusban megjelenő ismert szavak esetében. A legtipikusabb hiba a főnevek, melléknevek és igék összetévesztése, amely szinte minden esetben végzetes következményekkel jár az elemzés egészére. Erre látható példa a 4. táblázatban.

Mivel mind a szintaktikai, mind a függőségi elemzés ilyen félrevezető információkon alapul, a hiba továbbterjed a rendszerben és az 5. táblázatban látható hibákat eredményez.

4. táblázat. Példamondat IV.

100/CD million/CD **sound/NN** good/JJ to/TO me/PRP ./.
For/IN airline/NN personnel/NNS ,/, we/PRP **cash/NN** personal/JJ **checks/VBZ** up/RP to/TO / 100/CD ./.

5. táblázat. Példamondat V.

-/: 100/CD million/CD sound/NN good/JJ to/TO me/PRP ./.
For/IN airline/NN personnel/NNS ,/, we/PRP cash/NN personal/JJ checks/VBZ up/RP to/TO $/$ 100/CD ./.
-/: me/PRP_to/TO xxx/xxx 100/CD million/CD sound/NN good/JJ ./.
airline/NN personnel/NNS_For/IN ,/, cash/NN personal/JJ up/RP_checks/VBZ_we/PRP 100/CD_$/$_to/TO ./.

5. A MOSES keretrendszer

A statisztikai gépi fordítás területén legelterjedtebben a frázisalapú fordítást végző nyílt forráskódú MOSES nevű keretrendszert [7] használják, amely mind a tanítás, mind a dekódolás feladatára megoldást jelent. Mindemellett tartalmaz olyan segédprogramokat is, amelyek a nyelvmodellépítést és az automatikus kiértékelést is elvégzik. Ezt használtuk az itt leírt rendszerünk létrehozásához.

A MOSES alkalmas arra, hogy úgynevezett faktoros fordítórendszert hozzunk létre benne. A faktoros fordításba lehet a szövegben szereplő szavak puszta alakjánál mélyebb legalábbis morfoszintaktikai szintű információt belevinni. A fordítási faktorok olyanok lehetnek, mint a szóalakok felszíni alakja, töve alapvető szófaja, morfoszintaktikai jegyei. Faktoros fordítás esetén a keretrendszerben több fordítási, generálási és kontextuális nyelvmodellt hozhatunk létre, amelyeknek valamilyen kombinációját használja a rendszer, és így elvben képes lehet a korlátozottan rendelkezésre álló hiányos nyelvi adatok alapján is a sima szóalak alapú alaprendszernél jobb fordítások létrehozására olyan esetekben, ahol némi absztrakcióra van szükség az adott fordítás létrehozásához, mert pontosan azokat a szavakat nem látta a rendszer a tanítóanyagban, amelyekre az adott fordításhoz szükség lenne.

Sajnos azt találtuk, hogy a *MOSES-ben jelenleg létező konkrét faktorosfordítás-implementáció igazán nem alkalmas arra, hogy a magyarhoz hasonlóan gazdag morfológiájú nyelvekhez a rendelkezésünkre álló véges tanítóhalmaz alapján a sima szóalak alapú alaprendszernél jobb fordításokat hozzon létre. Ezért az itt leírt kísérleteink során egy alternatív megoldást próbáltunk létrehozni a morfológiai gazdagság és a két nyelv szerkezeteinek eltérő jellegéből adódó problémák (kötött morfémák a magyarban vs. izoláló szószerkezetek az angolban) kezelésére: morfémaalapú fordítót hoztunk létre.

6. Eredmények

Írott szövegek fordítása emberi olvasatra készül, ezért minden fordításnak a célja az, hogy emberek számára olvasható, érthető, az eredeti szöveggel azonos tartalmú fordítást hozzon létre. Mivel azonban az emberi kiértékelés lassú és drága, ezért elterjedt módszer a gépi fordítás minőségének vizsgálatakor annak automatikus kiértékelése, melyre több metrika is létezik. Ezek mindegyikének alapja az, hogy a géppel létrehozott fordítási eredményt ember által létrehozott referenciafordításhoz hasonlítják. Bár mindegyik metrikának vannak erősségei, és különböző szempontokat részesítenek előnyben a fordítás eredményének vizsgálatakor, önmagában egyik sincs mindig összhangban a fordítások emberi értékelésével. Munkánk során annak több fázisában végeztünk automatikus kiértékelést is a BLEU metrika szerint, de néhány esetet emberi kiértékeléssel is megvizsgáltunk, ami igazolta azt, hogy az automatikusan mért alacsonyabb értékek nem feltétlenül jelentenek rosszabb minőségű fordítást.

A rendelkezésünkre álló eredeti korpuszból a tanítás előtt félretettünk háromszor 1000 mondatból álló halmazokat a kiértékeléshez. Ezen kívül további vizsgálatokat végeztünk olyan teszthalmazon, amely a tanítóanyagban nem szereplő stílusú és témájú szövegeket (híreket) tartalmazott. Több rendszer eredményét mértük a különböző előfeldolgozási lépések hatásának értékelése céljából. Az alaprendszer a párhuzamos korpuszból minden előfeldolgozás nélkül tanított modell alapján fordított. A második fázis a morfémaalapú fordítás, ahol a forrásoldalon alkalmaztuk az elemzést és az átrendezést is, de a fordítás után a kapott morfémaalapú magyar mondatban nem generáltuk vissza a teljes magyar szavakat. Természetesen ebben az esetben sokkal magasabb BLEU-értéket kaptunk, de ez nem összehasonlítható a többi esettel, amelyekben szóalapú BLEU-értékeket kaptunk, így csak annak vizsgálatára alkalmas, hogy a morfémák milyen sikerrel kerültek bele a fordításba. A harmadik rendszer pedig a visszagenerált szövegen mért eredmény. Az egyes fázisok százalékban mért minőségét a 6. táblázat foglalja össze.

6. táblázat. A rendszerek eredményeit összefoglaló táblázat

	BLEU-érték		
Név	Baseline	Morf. elem. ford.	Generált ford.
test1	15,82%	64,14%	12,61%
test2	14,60%	57,39%	13,95%
test3	15,04%	57,84%	12,98%

Az eredményeken az látszik, hogy az alaprendszer BLEU-értéke a legmagasabb mindegyik teszthalmaz esetén. Ezek a különbségek azonban nem feltétlenül fejezik ki a valós minőségbeli, különbséget az egyes rendszerek által előállított fordítások között. Ennek oka, hogy a BLEU algoritmus minden eset-

ben egyszerűen a rendelkezésünkre álló egyetlen referenciafordítás szóalakjaihoz hasonlítja a rendszer által létrehozott fordítást. Mivel a hasonlításnál csak a szavak felszíni alakját veszi figyelembe, ezért teljesen mindegy, hogy egy egészen más, a fordításba egyáltalán nem illő szó került az eredménybe, vagy csak valamilyen ragozási hiba, esetleg szinonima szerepel.

A méréseknél kitűnik továbbá, hogy az egészen más témájú és stílusú hírkorpuszon is működik a rendszer, sőt ennek eredményére az átrendezési szabályok nagyobb hatással voltak, mint a tanítókorpuszhoz hasonló tesztek esetén.

6.1. Emberi kiértékelés

A különböző rendszerek által létrejött fordítások emberi vizsgálata során könnyen belátható, hogy a fent szereplő alacsony BLEU-értékek nincsenek összhangban a valódi minőséggel. A lefordított mondatok nagy része közelebb állt az eredeti mondat jelentéséhez az átrendezést és generálást alkalmazó rendszer esetén.

7. táblázat. Példamondat VI.

Eredeti mondat:	Nayla arrived then and the argument about the climb had begun .
Átrendezett mondat:	Naylum/[NNP] arrive/[VB] [Past] then/[RB] and/[CC] the/[DT] <zone> climb/[NN] about/[IN] </zone> xxx/[xxx] argument/[NN] <zone> begin/[VB] [PPart] have/[VB] [Past] </zone> ./[.]
Morféma alapú magyar fordítás:	Nayla/[FN] odaér/[IGE] [Past] [e3] ,/[PUNCT] a/[DET] haladás/[FN] [DEL] vitatkozik/[IGE] [INF] kezd/[IGE] [Past] [e3] ./[PUNCT]
Generált magyar:	Nayla odaért , a haladásról vitatkozni kezdett .
Baseline fordítás:	nayla odaért , és az a mászni kezdett .

6.2. Hibajelenségek

Az automatikus kiértékelő módszer hiányosságai mellett számos egyéb, a későbbiekben javítható probléma is megfigyelhető a fordítás minőségének ellenőrzésekor.

- Az angol szófajiegyértelműsítő-rendszer hibái: ha egy szó rossz szófaji címkét kap a fordítandó mondatban, akkor mivel a tanítás során a fordítási modellben a helyes címke a gyakoribb (ami kellően nagy korpusz esetén elvárható), ezt az alakot nem fogja tudni lefordítani még akkor sem, ha egyébként a szó önmagában gyakran előfordul. Ugyanakkor előfordulhat az is, hogy egy szónak többféle szófajú fordítása is szerepel a fordítási modellben, melyek közül a szövegkörnyezettől függően több is lehet helyes. Ezért ha az aktuálisan fordítandó mondatban rossz címke szerepel, akkor az annak megfelelő hibás fordítás kerül az eredménybe.

- Szintén az angol forrásszöveg hibás elemzése okozhat olyan hibát, ami a függőségi relációknál jelenik meg, így az átrendezési szabályok is helytelenül

8. táblázat. Példamondat VII.

Eredeti mondat:	For 50 years , barely a whisper .
Átrendezett mondat:	50/[CD] <zone> year/[NN] [PL] For/[IN] </zone> ,/[,] barely/[RB] a/[DT] whisper/[VB] ./[.]
Morféma alapú magyar fordítás:	50/[SZN_DIGIT] év/[FN] [PL] [TER] ,/[PUNCT] alig/[HA] egy/[DET] suttog/[IGE] [e3] ./[PUNCT]
Generált magyar:	50 évekig , alig egy suttog .
Baseline fordítás:	50 éve , alig egy suttogás .

hajtódnak végre. Ekkor olyan kifejezések kerülhetnek rossz helyre, melyek eredeti állapotukban jobbak voltak, s helyes elemzés esetén ott is maradtak volna. A Sinbad nem tulajdonnévként, ezzel szemben a valójában számnév Thousand tulajdonnévként címkézése alapvetően hibás szintaktikai elemzéshez vezetett, amelynek következtében az Ezeregy éjszaka fordítása (és Szindbádé is) zátonyra futott.

- Mivel a dekódolás során az egy szóhoz tartozó, de külön egységként megjelenő morfémákat bár külön tokenként, de egy egységként kezeltük (zónák), a fordítás során az ezeken átívelő frázisok nem érvényesültek. A zónahatárok lazább kezelése megoldhatná ezt a problémát.

- A tanító és a tesztkorpuszok minősége jelentős mértékben befolyásolja a fordítás minőségét is. Ez nemcsak amiatt jelent problémát, hogy bizonyos kifejezéseket hibásan tanul meg, hanem az automatikus kiértékelés során is sokszor hibás referenciafordításhoz végzi a hasonlítást. Ezért bár az eredeti mondat fordításának megfelel a létrejött fordítás is, ezekben az esetekben semmiképpen nem hasonlítható a referenciához.

- Mivel a fordítás során a fordítandó kifejezések alapegységei a morfémák, ezért ezek előfordulhatnak rossz szó mellé kerülve is, hiszen ugyanaz a toldalékmorféma egy mondaton belül többször is előfordul, a fordítási modell pedig több, az adott mondatban akár nem megfelelő szóhoz is hozzákapcsolhatja ezeket. Így a generálás során a toldalékok nem feltétlenül kerülnek a megfelelő szóra, illetve a kívánt helyen nem jelennek meg. Úgy látjuk, hogy a morfémaalapú fordítás alapvető problémát jelent már a tanító anyagban szereplő szóösszerendelések (illetve a mi esetünkben morféma-összerendelések) számára is, amelyek alapján a fordítóban használt frázistábla készül, ugyanis a hosszabb mondatokban ugyanaz a funkcionális morféma számos példányban előfordulhat, és a rendszerben használt Giza++ szópárosító algoritmus ezeket nem jól párosítja össze.

7. Összegzés

Cikkünkben bemutattunk egy olyan frázisalapú angol-magyar nyelvpárra készült hibrid fordító rendszert, melyet az automatikus statisztikai modellek használata mellett elő- és utófeldolgozási lépésekkel egészítettünk ki. Ezeknek célja az angol nyelvű mondatok átalakítása a magyarhoz jobban hasonlító szerkezetekké.

Ezekkel a transzformációkkal sikerült olyan fordításokat létrehozni, melyeknek bár az automatikus kiértékelés során mért minősége nem javult az alaprendszerhez képest, emberi olvasatra mégis sokszor sokkal jobbak annál. Számos olyan jelenség helyesen fordítható ezzel a módszerrel, melyet a hagyományos statisztikai gépifordító-rendszer nem tud kezelni. Bemutattuk azokat a hibajelenségeket is, melyeknek megoldása a további terveink része, s ezen kritikus pontok feloldása után további javulást várhatunk, ami jelentős áttörést jelentene az angol-magyar gépi fordítás területén.

Köszönetnyilvánítás

Ez a projekt a TÁMOP: 4.2.1.B – 11/2/KMR-2011–0002, valamint a MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport támogatásával készült. Továbbá köszönetet szeretnénk mondani Orosz György kollégánknak segítségért.

Hivatkozások

1. Birch, A., Osborne, M., Koehn, P.: Predicting Success in Machine Translation. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (2008) 745–754
2. Siklósi, B., Prószéky, G.: Statisztikai gépi fordítás eredményének javítása morfológiai elemzés alkalmazásával (2009) Msc diplomaterv.
3. Halácsy, P., Kornai, A., Németh, L., Sass, B., Varga, D., Váradi, T., Vonyó, A.: A hunglish korpusz és szótár. In: III. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2005) 134–142
4. Orosz, G., Novák, A.: Purepos – an open source morphological disambiguator. In: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science., Wroclaw, Poland (2012)
5. Prószéky, G., Kis, B.: A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. ACL '99, Stroudsburg, PA, USA, Association for Computational Linguistics (1999) 261–268
6. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: LREC-06. (2006)
7. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the ACL 2007 Demo and Poster Sessions, Prague, Association for Computational Linguistics (2007) 177–180

III. Korpusznyelvészet

Nyelvtanfejlesztés, implementálás és korpuszépítés: A HunGram 2.0 és a HG-1 Treebank legfontosabb jellemzői

Laczkó Tibor, Rákosi György, Tóth Ágoston, Csernyi Gábor

Debreceni Egyetem, Angol-Amerikai Intézet
4032 Debrecen, Egyetem tér 1.
{laczko.tibor, rakosi.gyorgy, toth.agoston,
gabor.csernyi}@arts.unideb.hu

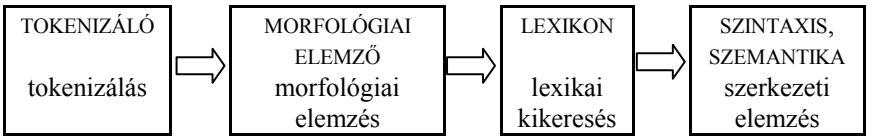
Kivonat: Cikkünkben beszámolunk kutatócsoportunk komplex elméleti nyelvészeti és implementációs vállalkozásának eddig elért eredményeiről. A kutatócsoport alapvető célja a magyar nyelv egy lehetséges generatív nyelvészeti modelljének a kidolgozása és ennek a modellnek az implementációja. Az elméleti keret a Lexikai-Funkcionális Grammatika, az implementációs platform a *Xerox Linguistic Environment*. Ebben a nagyobb ívű nyelvelméleti és nyelvtechnológiai kutatási folyamatban egy fajsúlyos részfeladatot is vállaltunk egy projekt keretében: egy jelentős méretű és sokféle felhasználási lehetőséget biztosító treebank létrehozását. A cikkben röviden jellemezzük az általános megközelítési keretünket, majd bemutatjuk azt, hogy ebbe hogyan ágyazódik bele a treebankes projekt: milyen nyelvelméleti és implementációs kihívásokkal kellett szembenéznünk, és milyen megoldásokat alkalmaztunk. Ezután részletesen tárgyaljuk és szemléltetjük a treebank legfőbb jellemzőit.

1 Bevezetés

Lexikai-Funkcionális Grammatikai Kutatócsoportunk (http://hungram.unideb.hu) 2008-ban egy OTKA projekt keretében kezdte el egy Lexikai-Funkcionális Grammatika (LFG) alapú magyar nyelvtan kidolgozását és – ezzel párhuzamosan – a nyelvtannak az implementálását (HunGram 1.0) az XLE (*Xerox Linguistic Environment*) platformon (a további részleteket l. a 3. szekcióban). Egy későbbi (de az előzővel párhuzamosan futó), TÁMOP projekt keretében egy másfél millió szavas magyar treebank összeállítását vállaltuk, amelynek a „rendező elve és motorja” a másik implementációs nyelvtannak egy olyan „átfejlesztése”, amely a treebank céljait közvetlenebbül és eredményesebben szolgálja. (A két projekt pályázati részleteit l. a *Köszönetnyilvánítás* szekcióban.) Cikkünkben egyrészt bemutatjuk ezt a treebankes nyelvtanváltozatot (a két nyelvtan közötti legfontosabb hasonlóságok és eltérések kiemelésével és illusztrálásával), másrészt beszámolunk az elvégzett korpuszfejlesztési munkálatokról (célok, eszközök, eredmények).

2 A HunGram nyelvtan automatikus elemzésre szánt változata

A HunGram 1.0 moduláris felépítése (az XLE-s nyelvtanok mintájára, vö. [1] és [8]), a következő.



1. ábra: a HunGram 1.0 fő komponensei.

A tokenizáló az adott szófüzért a magyar nyelv sajátosságainak megfelelő tokenekre bontja. Ezek szolgálnak bemenetül a morfológiai komponens számára, amely egy véges állapotú átalakító (*finite state transducer: fst*). A mi nyelvtanunk fst-je például a *játékot* főnevet és az *ették* igét a következő címkékkel (az angol eredeti alapján: *tag*ekkel) jellemzi.

(1) a. játék "+Noun" "+Sg" "+Acc"
b. eszik "+Verb" "+Past" "+Def" "+Pl" "+3P"

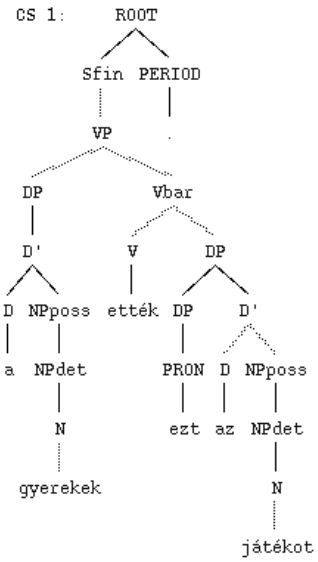
A lexikonban nemcsak szavaknak vannak megfelelő jellemzéssel ellátott tételei, hanem a különböző morfoszintaktikai jegyeket hordozó címkéknek is. Ezek lexikai tételeiben alapvetően funkcionális egyenlőségek révén rögzítjük az általuk kifejezett morfoszintaktikai jellegű információkat. Például a +Acc és a +Pl címkéknek a mi nyelvtanunk lexikonjában – egyebek között – a következő tételei találhatók.

(2) a. "+Acc" N_SFX XLE (↑ CASE)= acc
b. "+Pl" (i) N_SFX XLE (↑ NUM)= pl
(ii) V_SFX XLE (↑ SUBJ NUM)= pl

(2a) tanúsága szerint a +Acc címke főnévi tövekhez kapcsolódik, és azt a morfoszintaktikai információt kódolja, hogy a főnév (főnévi csoport) esetjegye akkuzatívusz. (2b) pedig azt mutatja, hogy egy címke különböző morfoszintaktikai információkat hordozhat attól függően, hogy milyen tőhöz járul. Egy főnévi tő esetében a főnév többességét jelöli, egy igei tő esetében viszont az alany többességét kódolja.

A szintaktikai elemzés alapját – az LFG felépítésének és elveinek megfelelően – egy olyan frázisstruktúrás szabályrendszer nyújtja, amelyben az egyes csomópontok szimbólumai megfelelő funkcionális annotációkkal is el vannak látva. Ilyen módon tud az LFG és (ebből következően) az XLE-ben implementált változata két parallel szintaktikai reprezentációt rendelni minden egyes jól megformált mondathoz: egy összetevős szerkezetet és egy funkcionális szerkezetet. Például a HunGram 1.0-ban a (3)-beli mondat egyik lehetséges összetevős szerkezetét a 2. ábra és az ennek megfelelő funkcionális szerkezetet a 3. ábra szemlélteti. Az összetevős szerkezet alapvetően a mondat kategoriális és szórendi jellemzőit ragadja meg, míg a funkcionális szerkezet a grammatikai viszonyokat ábrázolja (a grammatikai funkciókat és a releváns morfoszintaktikai információkat).

(3) *A gyerekekették ezt a játékot.*



2. ábra: *A gyerekek ették ezt a játékot* mondat összetevős szerkezete a HunGram 1.0-ban.

```
[PRED      'eszik<[2:gyerek], [95:játék]>'
 SUBJ      [PRED  'gyerek'
            NTYPE [NSEM [COMMON +]
                   NSYN common]
          2[CASE nom, DEF +, NUM pl, PERS 3]]
 OBJ       [PRED   'játék'
            NTYPE [NSEM [COMMON +]
                   NSYN common]
            SPEC 76[PRED 'pro'
                    CASE acc, DEIXIS proximal, NUM sg, PERS 3, PRON-TYPE demon]
          95[CASE acc, DEF +, NUM sg, PERS 3]]
 FOCUS     [2:gyerek]
 TNS-ASP   [MOOD indicative, TENSE past]
45[STMT-TYPE decl]
```

3. ábra: *A gyerekek ették ezt a játékot* mondat funkcionális szerkezete a HunGram 1.0-ban.

A HunGram 2.0-nak a HunGram 1.0-ból való kifejlesztése során a legfontosabb célunk az volt, hogy egy a magyar mondatokat automatikusan, a lehető legkevesebb többértelműséggel feldolgozni képes mondattani elemzőt hozzunk létre. Másrészt ugyanakkor alapvető kívánalom volt ezzel a nyelvtanváltozattal szemben is, hogy nyelvészeti szempontból is teljes és megbízható elemzéseket adjon. Ennek megfelelő-

en egy pusztán és szigorúan *csak nyelvészeti* megfontolások alapján szerkesztett nyelvtanváltozathoz képest sekélyebb, de a gépi feldolgozás kontextusában értelmezett valódi sekély nyelvtanoknál jóval gazdagabb nyelvtanváltozatot hoztunk létre. A nyelvtan hatékonyságát azáltal is igyekeztünk növelni, hogy az egyes köztes nyelvtanállapotok szerint leelemzett munkakorpuszból vett véletlenszerű elemzési minták helyességét több körben is manuálisan ellenőriztük, majd az észlelt hiányosságokat folyamatosan kiigazítottuk magában a nyelvtanban. A HunGram 2.0 így egy több szempontból is kiérlelt nyelvtanváltozatnak tekinthető.

A fenti követelmények jellege miatt a HunGram 2.0 elsősorban a pontosságra és kevésbé a lefedettségre törekszik. A következőkben röviden áttekintjük azokat a főbb tervezési jegyeket, amelyek ennek a közvetlen célnak a megvalósulását segítették elő. Az illusztrációként idézett mondatok forrása minden esetben maga a HG-1 Treebank.

A HunGram 2.0 szabályai nem generálnak pusztán nyelvészeti szempontból érdekes kétértelműségeket. Például az alábbi mondatban szereplő birtokos szerkezetben a határozott névelő elvileg tartozhat magához a birtokos főnévi csoporthoz (*a dzsungel*) vagy a teljes (félkövérrel szedett) birtokos szerkezethez:

(4) ***A dzsungel könyve*** *leszámol egy illúzióval*.

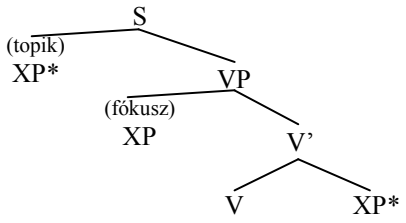
Ezt a fajta, inkább elméleti, mint gyakorlati jelentőségű kétértelműséget egyszerűen kizártuk azáltal, hogy a megfelelően megszorított nyelvtanunk csak a birtokos szerkezet egészéhez tudja hozzárendelni a határozott névelőt.

Általában véve olyan mögöttes mondattant alkalmazunk a HunGram 2.0 nyelvtanváltozatban, amelyet kimerítően meghatároznak a morfológiai és a sorrendi információk. Így nincs például mondattanilag kódolt információs szerkezetünk (nincs pl. fókusz, topik vagy kontrasztív topik), mert ennek pontos beazonosításához a fenti információk *gyakran* nem elégségesek. Vegyük például azt a feladatot, hogy egy az igét megelőző főnévi csoportról el kell dönteni, fókusz-e vagy topik (a korábbi (3) mondatot (7)-ként idézzük újra).

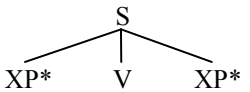
(5) ***Zoli*** *megette a radíromat*! **topik**
(6) *A sárkányok* ***csak a szüzeket*** *eszik meg, vagy rosszul tudom*? **fókusz**
(7) ***A gyerekek*** *ették ezt a játékot*. **topik/fókusz**
(8) *Az elmúlt század közepéig* ***a japánok*** *nem ettek marhahúst*. **topik/fókusz**

(5)-ben az igekötő ige előtti pozíciója kizárja a félkövér főnévi csoport fókuszos elemzését, (6)-ban viszont az igekötő posztverbális helyzete és a *csak* partikula jelenléte szükségszerűvé teszi azt. A HunGram 1.0 érzékeny is ezekre a szintaktikai információkra, és az ilyen szerkezetek esetén megbízhatóan el tudja dönteni, hogy lehet-e egy adott főnévi csoport a tagmondat fókusza vagy sem. (7) és (8) esetében viszont semmilyen morfoszintaktikai információ nem áll rendelkezésre, hogy el tudjuk dönteni, fókusz-e vagy topik a preverbális főnévi csoport. A HunGram 1.0 ilyenkor egy fókuszos és egy topikos elemzést is generál. Mivel az ilyen esetek igen gyakoriak, és az itt ismertetett megfontolásokhoz hasonlóak más esetekre is érvényesek, a HunGram 2.0-ban az ilyen kétértelműségek kiküszöbölése érdekében egy alapvetően lapos, az információszerkezetet nem grammatikalizáló mondatszerkezetet tételezünk föl. A HunGram 1.0-ban a számunkra most releváns, meghatározó mondatszerkezeti váz a (9a) mintázatát követi, míg a HunGram 2.0-ban a (9b) az alapvető rendező elv.

(9) a.



b.



A kategoriális vagy morfoszintaktikai jegyek miatti kétértelműségek egy részét igyekszünk gyakorisági megfontolások alapján már a lexikonban megszorítani. Tipikus példa az ilyen kétértelműségekre a melléknévként is lexikalizálódott melléknévi igenevek esete. A *borzasztó* szót például melléknévi lemmaként vettük fel a szótárban, letiltva egyúttal az (*el*)*borzaszt* ige folyamatos melléknévi igenévi használatát. Ennek következtében az alábbi mondatban a félkövérrel szedett szót csak melléknévként elemzi a nyelvtan, igenévként nem.

(10) *Hát* ***borzasztó*** *nevet választottál.*

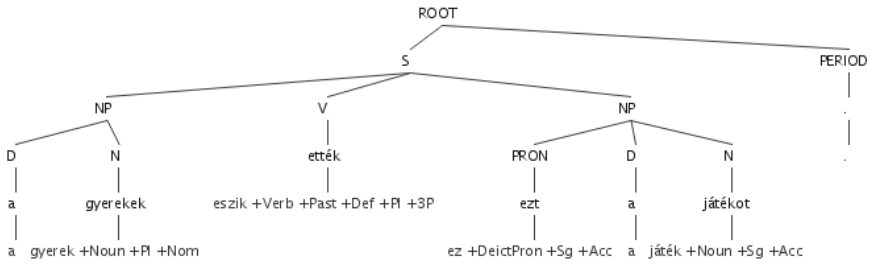
Mivel a szó valódi igenévi használata viszonylag ritka (vö. *a Jánost elborzasztó név*), többet nyerünk a potenciális kétértelműség kizárásával, mint a fenntartásával.

Végezetül egyes valós (vagyis akár jelentésbeli különbségekkel is járó) szintaktikai kétértelműségeket kizárunk a HunGram 2.0-ban. Például a HunGram 1.0-ban az általános LFG-s elméleti és XLE-s implementációs rendszerek felépítésének megfelelően az elemzőnk *összetevős szerkezeti* és *funkcionális szerkezeti* reprezentációt is ad. A funkcionális szerkezeti ábrázolásban igen gyakran van többszörös elemzés, és ennek az egyik rendszerszerű oka az, hogy bizonyos összetevőkhöz *oblikvuszi* és *adjunktumi* funkciót egyaránt képes a nyelvtan hozzárendelni. Vegyük például az alábbi mondatokat!

(11) *Képeket majd küldök jövő héten.*
(12) *Küldd a hírt a barátodnak!*

(12) datívuszos kifejezését szokás a *küld* ige oblikvuszi argumentumaként kezelni. Mivel azonban (11)-ben ez az összetevő nem szerepel, a (12)-ben található háromargumentumú *küld* tétel mellett mindenképpen fel kell venni a lexikonban egy kétargumentumú *küld* tételt is, amelynek csak alanyi és tárgyi argumentuma van. Emiatt viszont rögtön két elemzése is lesz (12)-nek, hiszen egyéb megszorítások híján elvileg egyaránt lehet benne akár a kétargumentumú vagy akár a háromargumentumú tétel is. Előbbi esetben a datívuszos kifejezés adjunktum, az utóbbiban oblikvuszi argumentum. A jelenlegi treebankünk szempontjából ez azért nem közvetlen gond, mert csak összetevős szerkezeti ábrázolást ad. Ugyanakkor a tervezett továbbfejlesztésre és a funkcionális szerkezeti ábrázolás beépítésére előre is gondolva a HunGram 2.0-ban az alkalmazható grammatikai funkciók közül kiiktattuk az oblikvuszt. Egy ilyen lépés *elméleti* motivációit részletesebben taglalja [5].

A HG-1 Treebankben tehát csak az összetevős szerkezeti elemzések jeleníthetők meg. (7) treebankbeli elemzését például az alábbi ábra mutatja (ugyanennek a mondatnak a HunGram 1.0 által generált összetevős szerkezetét a 2. ábrán láthattuk).



4. ábra: *A gyerekek ették ezt a játékot* mondat elemzése a HunGram 2.0-ban.

A szerkezetet generáló teljes nyelvtan bemutatására itt most nincs mód. Egy rövid átfogó ismertetés a treebank weboldalán érhető el (http://corpus.hungram.unideb.hu), a nyelvtant részletesebben ismertető publikációk jegyzéke pedig a kutatócsoportunk honlapján áll az érdeklődők rendelkezésére (http://hungram.unideb.hu).

Hadd álljon itt most csak egy rövid példa a nyelvtanfejlesztés során alkalmazott stratégiáink jellemzésére. A magyar nyelv egyik legnagyobb kihívást megtestesítő jelensége nyelvelméleti és implementációs szempontból egyaránt az igekötős igék viselkedése. A HunGram 1.0-ban [2] alapján kifejlesztettünk egy olyan megközelítést, amely az igekötős igék produktív és improduktív használatát is kielégítően és elvszerűen kezeli. Ennek részleteit [4] és [6] mutatja be. Tekintettel arra azonban, hogy ez a kezelésmód az improduktív használat esetében egyedi lexikai tételek bevitelét igényli, a produktív használat esetén pedig komplex frázisstruktúrás-annotációs mechanizmust, az automatizált elemzésre törekvő HunGram 2.0-ba ezt nem tudtuk átemelni. Ez utóbbi nyelvtanváltozatban lényegében – meglehetősen leegyszerűsített módon – közönséges határozói kategóriájúnak tekintünk minden igekötőt. Ezt annál is inkább könnyen megtehetjük, mert ez a nyelvtanváltozat egy olyan treebanket szolgál ki, amelyben nincs funkcionális szerkezeti reprezentáció. Márpedig az igazi kihívások ebben a dimenzióban jelentkeznek (grammatikai funkciók, argumentumszerkezet stb.). Mivel a jövőbeni implementációs nyelvtani munkálataink egyik legfontosabb célkitűzése az lesz, hogy a két nyelvtanváltozat azon vonásait, amelyek a másik változat számára is használhatók és hatékonyak lehetnek, igyekezzünk kölcsönösen figyelembe venni és minél teljesebb mértékben beépíteni, az igekötős igék kezelése esetében – a fentiek alapján értelemszerűen – azt fogjuk megvizsgálni, hogy van-e mód arra, hogy a HunGram 1.0 apparátusából átemeljünk olyan elemeket, amelyek a lehető legautomatizáltabb elemzést biztosítják úgy, hogy az igekötők természetének legfontosabb jellemzőit is megragadják.

3 Implementációs környezet, treebankfejlesztés

A magyar nyelvtan(ok) kidolgozása közvetlenül kapcsolódik a nemzetközi ParGram együttműködéshez (*Parallel Grammar*, lásd: [1]), amelyben több nyelvhez (angol,

német, urdu, francia, japán stb.) készül LFG nyelvtan, rendszeres egyeztetések mellett. A magyar grammatika minden elemét – a többi ParGram projekthez hasonlóan – az XLE munkafelületen [8] implementáltuk, mondattani elemzésre annak LFG-elemzőjét használtuk. Az XLE-ben található eszközöket, beleértve annak LFG-elemzőjét, nagyon hatékonynak találtuk, a rendszer robosztus és gyors, figyelembe véve azt is, hogy az LFG formalizmus szerinti mondattani elemzés inherens módon időigényes, NP-teljes probléma. A rendszer kifejlesztői több módon is igyekeztek kezelni ezt a problémát [9], többek között kiemelték a polinomiális időben megoldható feladatokat, és ezek feldolgozásával kezdik az elemzést, az exponenciális időben végrehajtható részfeladatok megoldása ezután következik. Az XLE elemzőjének egy másik tulajdonsága, hogy a szerkezeti többértelműségeket „csomagolt" módon, részfákra lokalizálva kezeli, ezzel segítve az eredmény gyors vizuális értelmezését. Ez ugyanakkor jól mutatja, hogy az XLE környezet elsősorban a humán feldolgozásra finomhangolt munkakörnyezet, miközben a további számítógépes feldolgozás szempontjából (a következőkben ismertetett treebankfejlesztési projekt esetében is) további munkafázist kellett ahhoz beiktatni, hogy a kimenet az általunk várt formát öltse.

Kutatócsoportunk a saját fejlesztésű nyelvtan és az XLE parser segítségével létrehozott egy 1,5 millió szavas treebanket (HG-1 Treebank). A treebank 1,5 millió szót (több mint 280 000 magyar mondatot) tartalmaz, morfológiai és mondattani annotációval ellátva. A HunGram 2.0 nyelvtanváltozat alapján végzett elemzést az NIIF szuperszámítógépes szolgáltatására támaszkodva, teljesen automatizálva állítottuk elő.

A korpuszt elsősorban az elméleti hátteret adó nyelvészeti kutatómunka kézzelfoghatóvá és felhasználhatóvá tételére, eredményeinek disszeminálására hoztuk létre és publikáltuk, ugyanakkor a fejlesztés felhasználható a nyelvoktatás, nyelvtanulás területén, a lexikográfiában, valamint elméleti nyelvészeti kutatásokban. A projekt website-ja (http://corpus.hungram.unideb.hu/) tartalmaz egy online lekérdezési felületet, valamint egy részletes leírást, útmutatót az elemzésekben található alaktani és mondattani jellemzők értelmezéséhez. Kiemeljük azt is, hogy a kutatócsoportunk nyelvtanírási projektjéhez a korpuszfejlesztési alprogram folyamatos tesztelési lehetőséget és visszajelzést biztosít.

Az elemzéseket tartalmazó korpusz kialakításához nyersanyagként a Magyar Webkorpuszt [3] használtuk, mely magyar weblapokról gyűjtött, elemzés nélküli szövegeket tartalmaz. A Magyar Webkorpusz (készítői által) szűrt változatából elemeztettünk annyi szöveget, hogy az elemzett mondatokban lévő szavak száma a 1,5 millió szót elérje.

Az elemzőnk által előállított kimenetet feldolgozva a mondattani fák tárolását a Tiger-XML leírónyelv segítségével oldottuk meg, amely kiváló eszköz fák reprezentálására [7]. Egy ágrajz kódolása a gyökérelem kijelölésével indul, utána a terminális szimbólumok felsorolása következik, melynek során a lexikai egységekhez kapcsolódóan a szófajt, a lemmatizált alakot és a morfológia által visszaadott összes jegyet tároljuk. Ezt követi az összes többi csomópont leírása legalább 1-1 kapcsolódó él meghatározásával.

Az adataink hozzáférhetőségének és kereshetőségének biztosítására létrehoztunk egy online lekérdezési felületet, amely lehetővé teszi a keresést szóra vagy lemmára, valamint a keresési találatok szűrését morfológiai jegyekre és a keresett szót tartalma-

zó összetevőre. A találatokról mondatelemzés-lista készül, ahonnan egy elemzést kiválasztva megkapjuk a megfelelő összetevős szerkezetet ágrajz formájában.

A korpuszfejlesztési projekt szoftver-infrastruktúrájának jelentős részét házon belül fejlesztettük ki. Az elvégzett programozási feladataink a következők voltak:

1. Mondatok elemeztetése a készülő nyelvtannal feltöltött XLE elemzővel, és a kimenet rögzítése (alternatív elemzésekkel).
2. Az összes lehetséges elemzés összetevős szerkezetének a kibontása és tárolása. A korpuszt ettől a ponttól XML dokumentumban tároljuk (TigerXML formátumban).
3. Alkorpuszok kezelése:
    a. korpuszfájlok darabolása és egyesítése,
    b. indexelés, statisztikák készítése (faszélesség, famélység, szavak és mondatok száma).
4. On-line lekérdezési felület létrehozása a következő főbb funkciókkal:
    a. keresés szóra vagy lemmára,
    b. keresés szűrése morfológiai jegyekre és a keresett szót tartalmazó összetevőre (szűrés beállítása űrlap segítségével),
    c. a találatok megjelenítése,
    d. a találati listából kiválasztott mondatelemzés ágrajzának megjelenítése.

A HG-1 Treebankben szereplő mondatok és elemzéseik adatbázisban történő rögzítését és kereshetővé tételét egy több fázisból álló feldolgozás előzte meg. Az előkészítés egy jelentős részét az XLE által előállított mondatelemzések összegyűjtése, egy köztes adatszerkezetre hozása (egy adott programnyelven), majd XML-formátumra alakítása képezte. A kiinduló pont az volt, hogy a keretrendszer az elemzéseket – amelyeket a korábban megírt mondatfeldolgozási és elemzési szabályok alapján állít elő – kimenetként Prolog programozási nyelven kódolt reprezentációban adja meg. A munkafolyamat ezen szakaszához tartozott tehát többek között a kimeneti Prolog fájlok (egy mondat a hozzá tartozó elemzésekkel = egy fájl) szerkezeti és tartalmi értelmezése, majd feldolgozása (vagyis a megfelelő adatszerkezet implementálása és az elemzések adatainak ebben az adatszerkezetben való rögzítése). Ezt követően kerülhetett sor a feldolgozott elemzések közül a (fentebb is említett) szabályok által létrehozható duplikált elemzések kizárólag egyszeres letárolására, a (gráfelméleti szempontból) köröket tartalmazó, és emiatt hibás összetevős szerkezetek kiszűrésére, valamint az ezáltal esetlegesen előálló elemzés nélkül maradt mondatok kizárására. További részfeladat volt a feldolgozás miatt a későbbiek során fontosnak bizonyuló információk, mint például az összetevős szerkezet (mint fa) mélységének és szélességének megállapítása, valamint az elemzések zárójelezett reprezentáció formájában történő elkészítése és letárolása olyan formában, amely az általunk is használt – az elemzések vizualizációját végző – phpSyntaxTree alkalmazás működéséhez szükséges.

A Prolog kódban tárolt elemzések egyedi, köztes adatszerkezetre való leképezése után lehetővé vált azoknak XML-alapú adatbázisba való építése. Az XML formátuma a már korábbi treebank alapú nyelvészeti alkalmazásokban is gyakran használt TigerXML lett, a fentebb is említett adatok (fa mélysége, szélessége; zárójelezett reprezentáció) tárolásához szükséges szerkezeti kibővítésekkel.

Az XML adatbázis létrehozását követően a további feladatokat az határozta meg, hogy egy weben használható, online lekérdezőfelületen keresztül – akár összetett feltételeket is tartalmazó, ugyanakkor viszonylag gyors – kereséseket lehessen végrehajtani a treebankben. Mivel a vállalt célok között szerepelt a keresési feltételek szóalakok és lemmák formájában történő megadása, továbbá a keresési lehetőségek részét képezte a szűrési feltételek morfológiai jegyekre és domináló szintaktikai kategóriákra szűkítése is, egy több táblából álló, SQL-alapú relációs adatbázis tervezése, valamint egy, a TigerXML forrásból adott SQL adatbázis formátumra alakító program készítése vált indokolttá. Így került sor egy megfelelő szerkezetű MySQL adatbázis kidolgozására, amelynek során fontos szempont volt, hogy külön táblában legyenek letárolva a mondatok, azok elemzései, az elemzésekben előforduló szóalakok morfológiai elemzésükkel, valamint azok lemmái szófaji kategóriájukkal. A kereshetőség felgyorsítása céljából a táblák indexekkel lettek ellátva.

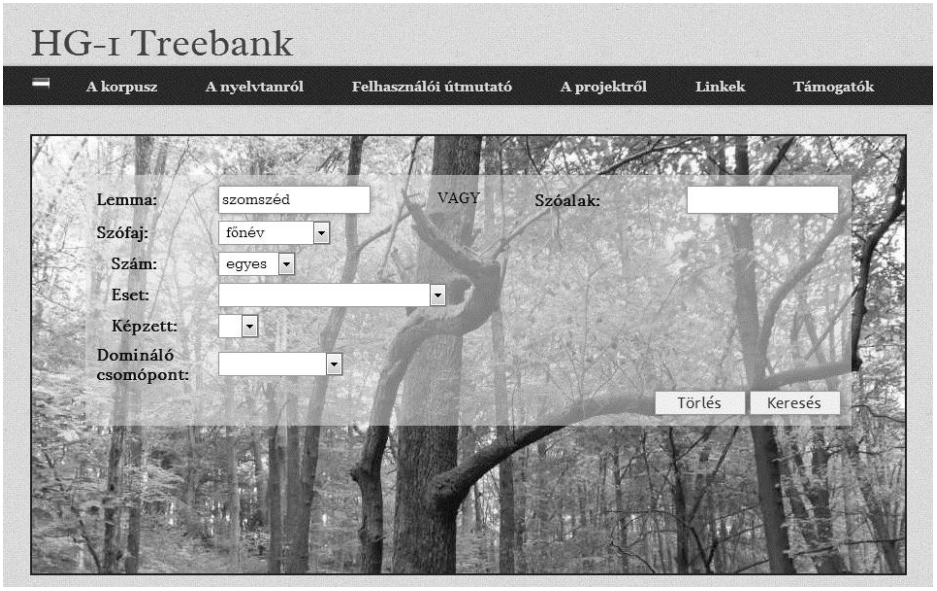
Az alkalmazás, amely a TigerXML forrás beolvasására, feldolgozására, és SQL-adatbázist töltő szkriptek írására lett kidolgozva, NIIF szuperszámítógépes környezetben került kipróbálásra és alkalmazásra. Ennek indokoltságát a relációs adatbázis jellegéből adódó, az integritás fenntartása érdekében végzett – a szóalakokra és lemmákra vonatkozó „szerepelt-e már”, „volt-e már” típusú – ellenőrzésekhez nélkülözhetetlen magas memóriaigény és hosszú, növekvő futási idő támasztotta alá.

A TigerXML forrásból ilyen módon létrehozott (MySQL) adatbázisból történő lekérdezésekhez ezt követően egy PHP-alapokon működő, weben elérhető interfész lett kialakítva, amelyben mint online űrlapban van lehetőség keresési kritériumok megadására. A keresendő adat minden esetben egy szóalak vagy lemma lehet, és további szűrési paraméterek is beállíthatók domináló szintaktikai csomópontok és morfológiai tulajdonságok/jegyek formájában. Ez utóbbiakat választómezőkön keresztül, opcionálisan lehet specifikálni. A keresési feltételeknek megfelelő eredményhalmazban megjelennek a mondatok és releváns elemzéseik is, az elemzések (mint összetevős szerkezetek) ágrajzos ábrázolására pedig a phpSyntaxTree (v1.10) alkalmazás került beépítésre az online keretrendszerbe.

Az interfész segítségével keresendő kifejezés minden esetben csak egy lemma vagy egy szóalak lehet. Ezek egyikének megadása kötelező, a keresés fő feltétele ezen alapszik. További szűrési kritériumként domináló csomópont bármilyen kereséshez beállítható, valamint szófaji kategória is megadható mint keresési feltétel. Ez utóbbi kiválasztása után (az adott szófajtól függő) további feltétel(ek)ként az egyéb morfológiai tulajdonságok szolgálhatnak. A szűrési feltételek szófaji kategóriánként csoportosított listája a következő:

- *főnév* (benne a tulajdonnevekkel): szám, eset, képzett-e;
- *határozószó*: képzett-e;
- *ige*: szám, idő, mód (feltételes/felszólító/kijelentő), műveltető-e, határozott-e, képzett-e;
- *igenév*: típus, szám, eset (csak melléknévi igenevek esetén), képzett-e;
- *kötőszó*;
- *melléknév*: szám, fok, eset, képzett-e;
- *névmás*: típus, szám, eset;
- *névutó*;
- *számnév*: szám, eset.

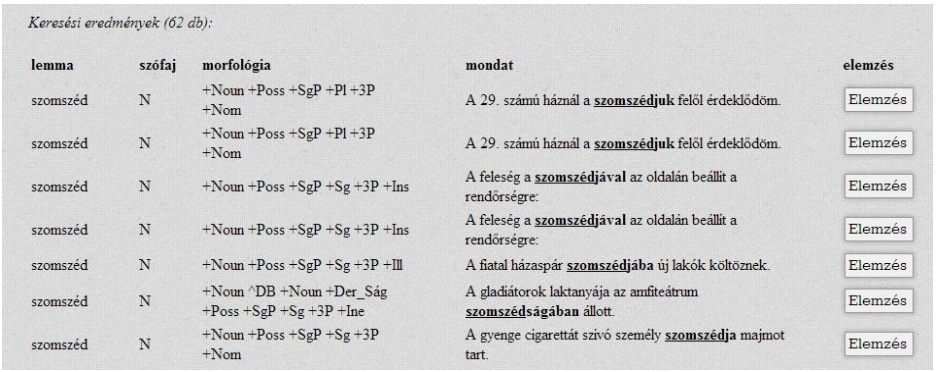
Az egy keresésre beállított szűrési opciók úgy működnek, hogy azok mindegyikének (egyszerre) teljesülnie kell a treebankben történő kereséskor. Az 5. ábra egy keresést szemléltet.



5. ábra: Keresés a *szomszéd* lemmára mint egyes számú főnévre.

A keresés eredményeit a lekérdezőfelület táblázatos formában jeleníti meg. Lemmára kereséskor listázásra kerül a lemma annak szófaji kategóriájával, azon szóalakoknak a morfológiai elemzése, amelyeknek az adott lemma tényleges lemmája, valamint a mondatok, amelyekben a lemma (bármilyen szóalakkal) előfordul. Abban az esetben, ha a fő szűrési paraméter szóalak, a táblázat értelemszerűen leszűkül a szófaji kategória oszloppal. A találatok számát a keresési mezők alatti szövegrész mutatja. A táblázatot képző eredménylistába a mondatok ábécésorrendben kerülnek.

A keresés találatainak áttekinthetőbbé tételét kiemelések is segítik (l. a 6. ábrát). Azon mondatokban, amelyekben egyszer fordul elő az adott keresési kifejezés (lemma vagy szóalak), lemmára kereséskor a mondat oszlopban aláhúzás jelöli a lemmát – amennyiben az morfo(fono)lógiai alternáció nélkül van jelen –, a szólak pedig félkövér betűstílussal jelenik meg, függetlenül attól, hogy lemmára vagy szóalakra szűrünk.



Keresési eredmények (62 db):

lemma	szófaj	morfológia	mondat	elemzés
szomszéd	N	+Noun +Poss +SgP +Pl +3P +Nom	A 29. számú háznál a **szomszédjuk** felől érdeklődöm.	Elemzés
szomszéd	N	+Noun +Poss +SgP +Pl +3P +Nom	A 29. számú háznál a **szomszédjuk** felől érdeklődöm.	Elemzés
szomszéd	N	+Noun +Poss +SgP +Sg +3P +Ins	A feleség a **szomszédjával** az oldalán beállít a rendőrségre:	Elemzés
szomszéd	N	+Noun +Poss +SgP +Sg +3P +Ins	A feleség a **szomszédjával** az oldalán beállít a rendőrségre:	Elemzés
szomszéd	N	+Noun +Poss +SgP +Sg +3P +Ill	A fiatal házaspár **szomszédjába** új lakók költöznek.	Elemzés
szomszéd	N	+Noun ^DB +Noun +Der_Ság +Poss +SgP +Sg +3P +Ine	A gladiátorok laktanyája az amfiteátrum **szomszédságában** állott.	Elemzés
szomszéd	N	+Noun +Poss +SgP +Sg +3P +Nom	A gyenge cigarettát szívó személy **szomszédja** majmot tart.	Elemzés

6. ábra: Találati lista kiemelésekkel a *szomszéd* lemmára mint egyes számú főnévre.

A találatok összetevős szerkezetének vizualizációja egy külön ablakban jelenik meg az adott mondat melletti „Elemzés” gombra kattintást követően. Egy ilyen példát mutatott be a 4. ábra. Az ágrajzban az alapértelmezés szerint külön színnel jelölt terminálisok szintjén a mondatban szereplő szóalakok morfológiai elemzései láthatók. (Ahol nem jelenik meg morfológiai elemzés, ott az a nyelvtanírás során már korábban felül lett írva LFG-formalizmuson alapuló lexikai tétellel.) Az összetevős szerkezetet szemléltető felület a rendszer sajátosságait kihasználva lehetőséget biztosít a megjelenítést kényelmesebbé tevő beállítások változatására is (pl. a betűméret változtatása, a terminálisok külön színnel való jelölése stb.), amely nagymértékben hozzájárul a Treebank komplex, integrált, ugyanakkor felhasználóbarát rendszeréhez.

A korpusz legfőbb célja és értéke a munkacsoport által kifejlesztett magyar LFG nyelvtan kézzelfoghatóvá és felhasználhatóvá tétele. Alkalmazható a nyelvoktatás és a nyelvtanulás területén – a korpuszalapú megoldások összes előnyével: motiváló autentikus élőnyelvi szövegekkel dolgozhatunk olyan módon, hogy a tanulás nyelvi felfedezéssé válik. Ugyancsak fontosak számunkra a lehetséges lexikográfiai alkalmazások, valamint a korpusz felhasználása elméleti nyelvészeti kutatásokban (melyre közvetlen példa saját nyelvtanírási projektünk is, amelyhez a korpusz folyamatos tesztelési lehetőséget és visszajelzést biztosított).

Köszönetnyilvánítás

A cikk elkészítését részben az OTKA K 72983 számú kutatási projekt, részben pedig a TÁMOP 4.2.1./B-09/1/KONV-2010-0007 számú projekt támogatta. A TÁMOP projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg. Laczkó Tibor és Rákosi György kutatásait a Magyar Tudományos Akadémiának a Debreceni Egyetemen működő Elméleti Nyelvészeti Kutatócsoportja is támogatta.

Hivatkozások

1. Butt, M., King, T.H., Niño, N., Segond, F.: A grammar writer's cookbook. CSLI Publications, Stanford (1999)
2. Forst, M., King, T.H., Laczkó, T.: Particle verbs in computational LFGs: Issues from English, German, and Hungarian. In: Miriam, B., King, T.H. (eds.): Proceedings of the LFG'10 Conference. CSLI Publications, Stanford (2010) 228–248
3. Kornai, A., Halácsy, P., Nagy, V., Oravecz, Cs., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In: Kilgarriff, A., Baroni, M. (eds.): Proceedings of the 2nd International Workshop on Web as Corpus ACL-06 (2006) 1–9
4. Laczkó, T., Rákosi, Gy.: On particularly predicative particles in Hungarian. In: Butt, M., King, T. H. (eds.): Proceedings of the LFG '11Conference. CSLI Publications, Stanford (2011) 299–319
   Online: http://cslipublications.stanford.edu/LFG/16/papers/lfg11laczkorakosi.pdf
5. Rákosi, Gy.: Non-core participant PPs are adjuncts. In: Butt, M., King, T. H. (eds.): Proceedings of the LFG '12Conference. CSLI Publications, Stanford (Megj.e.)
6. Rákosi, Gy., Laczkó, T.: Inflecting spatial particles and shadows of the past in Hungarian. In: Butt, M., King, T. H. (eds.): Proceedings of the LFG '11Conference. CSLI Publications, Stanford (2011) 440–460
   Online: http://cslipublications.stanford.edu/LFG/16/papers/lfg11rakosilaczko.pdf
7. The TIGER-XML treebank encoding format.
   http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html
8. XLE Documentation. http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html
9. XLE. http://www2.parc.com/isl/groups/nltt/xle/

HunLearner: a magyar nyelv nyelvtanulói korpusza

Vincze Veronika[1], Zsibrita János[2], Durst Péter[3], Szabó Martina Katalin[4]

[1] MTA-SZTE Mesterséges Intelligencia Kutatócsoport
vinczev@inf.u-szeged.hu
[2] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
zsibrita@inf.u-szeged.hu
[3] Szegedi Tudományegyetem, Hungarológia Központ
durst.peter@gmail.com
[4] Szegedi Tudományegyetem, Magyar Nyelvészeti Tanszék
szabomartinakatalin@gmail.com

Kivonat: Cikkünkben bemutatjuk a HunLearner korpuszt, mely a magyart mint idegen nyelvet tanulók által létrehozott szövegeket tartalmaz. A korpusz tartalmazza a morfológiailag hibás főnevek javított alakjait és a hiba kódját is. A javított alakok kézi annotációja lehetővé tette azt is, hogy megvizsgáljuk a hibák automatikus javításának lehetőségeit. Az eredmények azt mutatják, hogy már egyszerű módszerekkel is jelentősen lehet csökkenteni a hibás szóalakok számát egy nem sztenderd szövegben, ami ígéretesnek mutatkozik a nem sztenderd szövegek automatikus feldolgozására nézve.

1 Bevezetés

A magyar nyelvtechnológia eddig túlnyomórészt sztenderd magyar szövegek elemzésével foglalkozott, azonban számos olyan magyar nyelvű dokumentum létezik, amelynek sajátságai eltérnek a sztenderd nyelvtől. Közéjük tartoznak a webes szövegek, a nyelvjárási szövegek, illetve a magyart idegen nyelvként beszélők, továbbá az agysérültek vagy nyelvi zavarral rendelkezők által létrehozott nyelvi produktumok. Az ilyen jellegű szövegek feldolgozásához egyrészt a meglevő elemzők átalakítása, másrészről pedig annotált korpuszok létrehozása szükséges. Ennek első lépéseként az előadásban egy digitalizált magyar nyelvtanulói korpuszról számolunk be.

Nyelvtanulói korpuszoknak nevezzük azokat a korpuszokat, amelyek egy bizonyos nyelvet idegen nyelvként tanulók írott vagy hangzó szövegeit tartalmazzák (vö. [11]). Létrehozásuk célja, hogy fényt deríthessünk mindazokra a sajátságokra, amelyek a tanulók nyelvezetét (*köztes nyelv*, *interlanguage* [10]) az anyanyelvi beszélőkétől megkülönböztetik (vö. [7]). Mivel a digitalizált nyelvtanulói korpuszok lehetővé teszik a diákok nyelvi produktumainak alapos vizsgálatát, fontos szerepet tölthetnek be a kapcsolódó nyelvészeti kutatásokban, valamint az oktatási anyagok fejlesztésének folyamatában egyaránt. Emellett hathatós segítségül szolgálhatnak a hibakereső rendszerek értékelésében és fejlesztésében, valamint a lexikográfia területén a különböző szótárak, köztük az egynyelvű nyelvtanulói szótárak készítésében is (vö. [3,4,6]). Jelentős gyakorlati hasznuknak köszönhetően a nyelvtanulói korpuszok száma az

elmúlt években jelentősen megnövekedett, legtöbbjük azonban valamely nyugat-európai nyelv köztes nyelvi szövegeit tartalmazza [1]. A magyar nyelv vonatkozásában elmondható, hogy, bár a magyart idegen nyelvként tanulók nyelvi hibái régóta képezik vizsgálat tárgyát, a vonatkozó tanulmányok vizsgálati anyagaként nem digitálisan rögzített anyagokat használtak, és az adatok feldolgozása is manuálisan történt. Emellett a viszonylag kisméretű nyelvi anyagokat (10-20 válaszadó) általában a magyar és valamilyen másik nyelv kontrasztív elemzése alapján elemezték. Tudomásunk szerint ez idáig két olyan magyar nyelvtanulói korpusz készült, amelyet digitális formában dolgoztak fel: a BilingBank kínai–magyar, 11 interjút tartalmazó korpusz, valamint az Indiana Egyetem 14, egyenként 10-15 mondatból álló szöveget tartalmazó korpusza [4]. A HunLearner korpusz újdonsága abban rejlik a korábbiakhoz képest, hogy egyrészt jóval nagyobb méretű, mint az eddigiek, másrészt tartalmazza a morfológiailag hibás főnevek javított alakjait és a hibák kódját is.

2 Elméleti háttér és nemzetközi kitekintés

Bár a viszonylag csekély számú érintett miatt a magyar mint idegen nyelv tanítása soha nem foglalt el kitüntetett helyet a nemzetközi köztudatban, módszertana igen hosszú múltra tekint vissza és kiváló nyelvészek tevékenykedtek ezen a területen. A hazai nyelvészeti vizsgálódások ma is a korszerű nemzetközi kutatásokkal karöltve folynak, a magyar nyelv sajátosságainak figyelembevételével. Így nem hiányoznak az utóbbi évtizedek szakirodalmából a hibaelemzéssel foglalkozó tanulmányok sem, amelyek alapvetően a magyar nyelv tanulása és idegen nyelvként történő használata közben elkövetett hibákat[1] csoportosítják és elemzik.

Az elméleti háttér az utóbbi évtizedekben jelentősen megváltozott, hiszen az anyanyelv és az idegen nyelv részletes kontrasztív elemzésén alapuló, a hibákat előre megjósló és kerülni szándékozó behaviorista szempontú megközelítés helyett mára széles körben ismert és elfogadott fogalom lett a *köztes nyelv* (vö. 1. rész), amely a nyelvtanuló saját nyelvi rendszerére utal. Ebben a folyamatosan változó, szerencsés esetben a célnyelvhez egyre jobban közelítő rendszerben a hétköznapi értelemben vett hibákat a nyelvtanuló saját köztes nyelvének megnyilvánulásaként értelmezzük, amelyek a szabályalkotási folyamatokról tanúskodnak. Ennek megfelelően nem a tanulást akadályozó, zavaró jelenségekként szemléljük őket, hanem a nyelvtanulás folyamatának természetes és szükséges velejárójaként. Az anyanyelvet és a célnyelvet, valamint a köztes nyelv tulajdonságait egyaránt figyelembe vevő hibaelemzés tehát nagy segítséget nyújthat ma is a nyelvtanításban. A tanulói korpuszok számítógépes feldolgozásában a morfológiailag igen gazdag magyar nyelv számos kihívást támaszt, és bár már más finnugor nyelvek tanulói korpuszainak köszönhetően állnak rendelkezésre adatok [9], a hibák javítása és kódolása még ezekben a projektekben sem telje-

[1] A nyelvek tanulásának és elsajátításának vizsgálatakor lényeges feladat a célnyelvi szabályoknak nem megfelelő, rendszerszerű eltérések, azaz a valódi hibák (*error*), valamint a nyelvi szabályok tudásának ellenére, alkalmi jelleggel felbukkanó tévesztések (*mistake*) megkülönböztetése, mivel azonban a jelen tanulmány szempontjából ez a probléma nem releváns, a dolgozatban egységesen a *hiba* terminust használjuk.

sen megoldott. A közelmúlt nemzetközi eredményei inspirálóak: új nyelvtanulói korpuszok építéséből, annotálásából és a hibák kezeléséből álló komplex feladatokat sikerült már megoldani idegen nyelvként ritkábban tanított nyelvek esetében is (l. például a cseh nyelv nyelvtanulói korpuszát [8]). A HunLearner nyelvtanulói korpusz építésével arra törekszünk, hogy e hiányosságot a magyar nyelv vonatkozásában is pótoljuk.

3 A korpusz adatai

A HunLearner korpusz szövegei a Zágrábi Egyetem magyar szakos, horvát anyanyelvű hallgatóitól származnak. A diákok három témában írtak fogalmazást: (1) Nehézségek a magyar nyelv tanulásában; (2) Egy szimpatikus ember; (3) Egy Angliában dolgozó magyar levele a családjának. A fogalmazásokat számítógépen készítették el, amelyre legfeljebb egy óra állt a rendelkezésükre. A munka során szótárt, nyelvkönyvet, illetve internetes forrásokat nem volt szabad használniuk, emellett magyar billentyűzettel kellett dolgozniuk. A tényleges nyelvi anyagon kívül a válaszadókra vonatkozó adatokat is tárolunk, azaz a nyelvtanulók életkorára, nemére, anyanyelvére, egyéb idegen nyelvi ismeretére, a magyar nyelv tanulásával töltött eddigi időtartamra, valamint a célnyelvi országban eltöltött időre vonatkozó információkat. Mindezeket a későbbi elemzésekben szándékozzuk felhasználni. A korpusz főbb adatait az alábbi táblázat foglalja össze.

1. táblázat: A HunLearner korpusz adatai.

	Nehézségek	**Szimpatikus ember**	**Anglia**	**Összesen**
Szövegek száma	18	6	11	35
Mondatszám	559	134	258	951
Tokenszám	10433	1930	3936	16299

Az alábbiakban bemutatunk egy részletet a korpuszból:

Amikor én kisgyerek voltam minden évben apámmal Bosznában utaztam.Ott egy kis faluban megismertem egy öreg embert. A neve Bego volt. Ő nagyon erős volt és bőlscesz is. Amikor három fiatal ember földről nem tudhatott felhozni a fákat ő tudhatta. Egész napon tudhatott nehézek munkákat csinálni, erdőben egyedül fákat levágni,kecskékkelhegyekre sétálni és mindent enekelve és vakáció kivül csinált. Estén a háza előtt ült és gyrekeknek falúból ijedősök meséket elbeszél. Ha én ott is nyartam, minden estén a meséket is hallgattam. Nagyon szép volt ott maradni, mert Bego is tüzet megcsinált. Mindenki szeretti őt. Szomsedeinek mindenben segített és mindig mosolyos volt

4 Morfológiai hibák a korpuszban

A korpuszt a *magyarlanc* elemzővel [15] automatikusan elemeztük, majd az elemző által ismeretlennek minősített szavakat további elemzéseknek vetettük alá. Célunk a morfológiai hibák kategorizálása volt. Első lépésként a *hunspell* helyesírás-ellenőrző [12] segítségével javítottuk a hibásan írt szóalakokat. Azokban az esetekben ahol több lehetőséget is ajánlott a program, kézzel választottuk ki a kontextusba illőt. Ezzel a módszerrel az ismeretlen szavak 60%-ára kaptunk elemzést, a maradék 40% túlnyomó többsége idegen szó vagy tulajdonnév volt. Mivel jelenleg a főnévi hibák javítására koncentrálunk, kiszűrtük a főneveket (a javított szavak 45%-át), majd közülük is kiválasztottuk a morfológiai hibát tartalmazókat (azaz a szegmentálási hibát tartalmazó eseteket figyelmen kívül hagytuk). Így a további vizsgálataink alapját összesen 157 főnévi hibás szóalak képezte, ami a javított szavak közel 40%-át jelentette. A 2. táblázat bemutatja az ismeretlen, illetve a javított szavak korpuszbeli számát és arányát.

2. táblázat: Az ismeretlen, illetve javított szavak száma és aránya a korpuszban.

	Nehézségek	**Anglia**	**Szimpatikus ember**	**Összesen**
Szavak száma	8692	3271	1622	13585
Ismeretlen szavak (aránya)	393 (4,52%)	146 (4,46%)	128 (7,89%)	667 (4,91%)
A helyesírás-ellenőrző által felajánlott javítások	2328	614	679	3621
Az elfogadott javítások (aránya)	237 (60,31%)	110 (75,34%)	50 (39,06%)	397 (59,52)
A javított főnevek (aránya)	100 (42,19%)	58 (52,73%)	24 (48%)	182 (44,84%)
A kiszűrt főnevek (aránya)	80 (33,76%)	56 (50,91%)	21 (42%)	157 (39,55%)

Megjegyezzük, hogy a morfológiai elemző által ismeretlennek minősített szavak aránya jóval nagyobb a *Szimpatikus ember* alkorpuszban, mint a másik kettőben, és ugyanitt az elfogadott javítások aránya is jóval alulmarad a többi alkorpuszhoz képest. Ennek valószínűleg az lehet az oka, hogy a fogalmazások témájából fakadóan számos tulajdonnév, elsődlegesen személy- és helynév szerepel a szövegekben, amelyek elemzésére sem a *magyarlanc*, sem a *hunspell* nem volt képes.

A morfológiai hibák osztályozására egy saját kategóriarendszert és az ennek megfelelő kódrendszert hoztunk létre az általános nyelvtanári tapasztalat, valamint a magyar mint idegen nyelv vonatkozásában készült hibaelemzések alapján [5]. A következőkben az osztályozás részleteit mutatjuk be, példákkal illusztrálva a hibák egyes típusait.

A hibás szóalakoknál először is megvizsgáltuk, hogy a szótő vagy a toldalék-e a hibás (természetesen nem zártuk ki azt az esetet sem, hogy mind a kettő is tartalmazhat hibát egyszerre). A szótőben található hibákat aszerint bontottuk tovább, hogy többalakú tő nem megfelelő alakját tartalmazza-e a szó (pl. **kézem* a *kezem* helyett), illetve egyéb elírást, helyesírási hibát találhatunk benne (pl. **problámát* vs. *problémát*). A

szótő minőségét (helyes, hibás, utóbbi esetben mi a hiba jellege) a hibakódok első pozíciója kódolja.

A toldalékolással kapcsolatos hibákat alapvetően szintén két osztályra bontottuk (a két osztály szintén nem zárja ki egymást). Az első hibaosztály a hasonulással kapcsolatos hibákat foglalja magában, a második pedig a hangrenddel, kötőhangokkal és toldalékallomorfokkal kapcsolatos hibákat tartalmazza. A hibakód második pozíciója jelzi a hasonulási hibákat, a harmadik pozíció pedig a második toldalékolási hibaosztálynak feleltethető meg. A kód negyedik pozíciója azt tartalmazza, hogy egy vagy több morfémából áll-e a toldalék. A hibatípusok összefoglalása az alábbi táblázatban látható, példák segítségével illusztrálva.

3. táblázat: Hibatípusok.

Első pozíció – szótő	**Kód**	**Magyarázat**	**Példa**
	A	helyes	
	B	helyesírási hibát tartalmazó szótő	*problámát*
	C	többalakú tő nem megfelelő alakja	*kézek*
	X	egyéb hiba	
Második pozíció – hasonulás	1	nincs hasonulás és nem is kell	*kézt*, *kezet*
	2	van hasonulás, és jó, de egyéb probléma van a toldalékkal	*cukorram* (=*cukorral*)
	3	van hasonulás, de nem kellene	*hallak* (=*halnak*)
	4	nincs hasonulás, de kellene	*cukorval*
	5	van hasonulás, de hibás	*cukornal* (=*cukorral*)
	X	egyéb hasonulási hiba	
Harmadik pozíció – hangrend, kötőhangok, toldalékok allomorfjai	A	helyes allomorf	
	B	hangrendi hiba	*házben*
	C	rossz kötőhang	*házen* (=*házon*)
	D	fölösleges kötőhang	*söröt*
	E	hiányzó kötőhang	*templomt*
	F	fölösleges j birtokjel	*toldalékja*
	G	hiányzó j birtokjel	*kutyáa*
	H	hangrendi illeszkedés egyalakú toldaléknál	*éjfélker*
	X	egyéb toldalékolási hiba	
Negyedik pozíció – toldalékok száma	0	nincs toldalék	*problém*
	1	egy toldalék	*házben*
	2	egynél több toldalék	*kézemben*

A morfológiai hibák automatikus kódolására kifejlesztettünk egy szabályalapú rendszert, amely a hibás és helyes szóalak összevetése alapján rendeli hozzá a hibakódokat az egyes hibás szóalakokhoz. Az automatikus kódokat a *Nehézségek* alkorpuszon ellenőrizve azt állapítottuk meg, hogy azok minősége megfelel az elvárásoknak, 80 esetből mindössze 2 hibát találtunk.

Az alábbiakban bemutatunk egy mintát az automatikusan kódolt szóalakokból. A korpuszban szereplő alakot követi a javított szóalak, majd a hibakód következik:

viszonyot	*viszonyt*	*A1D1*
hidjai	*hídjai*	*C1A2*
rágozást	*ragozást*	*B1A1*
tanszékon	*tanszéken*	*A1C1*
gyokorlatokon	*gyakorlatokon*	*B1A2*

Az automatikus hibakódolás lehetővé tette az egyes hibatípusok számszerűsítését is. Ezáltal megvalósíthatóvá vált, hogy megállapítsuk a tő- és toldaléktévesztések arányát, illetve a hasonulási és hangrendi problémák arányát. A morfológiai jellegű hibák mellett automatikusan megvizsgáltuk az ékezettévesztéses hibák arányát is, hiszen a korpuszbeli szövegek előzetes tanulmányozása arra engedett következtetni, hogy az ékezetek helyes kitétele gyakori hibaforrás a nyelvtanulók körében. A mért adatokat a 4. táblázat foglalja össze.

4. táblázat: A morfológiai hibák száma a korpuszban.

helyesírási hibát tartalmazó szótő	122
többalakú tő nem megfelelő alakja	12
hangrendi hiba	5
rossz kötőhang	8
fölösleges kötőhang	3
hiányzó kötőhang	1
fölösleges j birtokjel	2
egyéb toldalékolási hiba	8
ékezet	40

Az eredmények szerint a leggyakoribb hibatípus a tőtévesztés (85%) volt, különös tekintettel az ékezetek nem megfelelő használatára (28%). A toldaléktévesztések közül pedig a hibás kötőhang volt a leggyakoribb (29%).

5 Az automatikus hibajavítás lehetőségei

A javított alakok kézi annotációja lehetővé teszi azt is, hogy megvizsgáljuk a hibák automatikus javításának lehetőségeit, így teszteltük néhány egyszerű módszer hatékonyságát a hibák kijavítására. Amennyiben a *hunspell* által javasolt első helyes szóalakot választottuk, akkor 81,86%-os pontosságot értünk el az összes javított szóalakot tekintve, ami az összes ismeretlen szóalak 49%-ának felel meg.

Ezen túl egy másik módszert is alkalmaztunk: megvizsgáltuk, hogy a *hunspell* által javasolt szóalakok közül melyek fordulnak elő a Szeged Treebankben [2], és, amennyiben több javasolt szóalak is szerepelt benne, a leggyakoribbat választottuk. Ez a módszer 83%-os pontosságot eredményezett, azonban csak 318 szó esetében tudtuk

alkalmazni, mivel az adatbázisban előfordultak olyan szóalakok, ahol a javítási javaslatok egyike sem szerepelt a korpuszban, így azokhoz nem tudtunk gyakoriságot hozzárendelni.

A fenti két megoldást végül kombináltuk egymással: első lépésben a leggyakoribb javasolt szóalakot rendeltük a hibás alakhoz, illetve azon szavak esetében, ahol ez nem volt lehetséges, a *hunspell* által javasolt első javított alakkal dolgoztunk. Ez a módszer végül 82,62%-os pontossághoz vezetett.

Eredményeink arra utalnak, hogy már egyszerű módszerekkel is jelentősen, körülbelül felére lehet csökkenteni a hibás szóalakok számát egy nem sztenderd szövegben, ami ígéretesnek mutatkozik a nem sztenderd szövegek automatikus feldolgozására nézve. További javítási lehetőségként a különféle tulajdonnévszótárak beépítése kínálkozik a morfológiai elemzőbe, különös tekintettel a nyelvtanulói korpusz szövegeit létrehozó tanulók nemzetiségére és földrajzi környezetére. A HunLearner esetében például egy horvát személy- és földrajzinév-szótár bizonyulna hasznosnak.

A korpuszban természetesen előfordulhatnak olyan esetek is, amikor a szóalak morfológiailag kifogástalan, azonban szintaktikailag nem illik a mondatba, mert például az ige más vonzatot kíván meg. Az ilyen esetek automatikus felderítése nem valósulhat meg pusztán morfológiai elemzés segítségével, ehelyett a szintaxishoz kell segítségért folyamodni. A korpuszt automatikus függőségi elemzésnek vetettük alá a *magyarlanc* 2.0 [15] függőségi moduljával, majd kinyertük belőle az igei vonzatkereteket. Összesen 953 vonzatkeret szerepel a korpuszban, melyeket összehasonlítottuk a Szeged Dependencia Treebankből [13] kigyűjtött vonzatkeretekkel [14], és amelyek nem szerepeltek benne (306 vonzatkeret, az összes keret 32,11%-a), azokat külön vizsgálat alá vetettük. Tekintve, hogy a magyarban nem kötelező fonológiailag megjeleníteni a névmási vonzatokat, kiszűrtük azokat az igéket, amelyek argumentumszerkezete üres volt, így 278 vonzatkeretet kaptunk (29,17%). Ezek közül 37 esetben az egyik vonzat ismeretlen vagy hibás szóalak szófaji kódot kapott, így a morfológiai elemzés tökéletlensége okán a szintaktikai elemzés sem lehetett kielégítő. Összesen tehát 241 olyan vonzatkeret (25,29%) található a korpuszban, amely további vizsgálatra szorul. Előzetes eredményeink szerint a problémás keretek egy része valóban hibás (pl. az *érdekel* ige részes esetű vonzattal: *nekem nem érdekel*), más esetekben a szintaktikai elemző hibázik, illetve lehetnek olyan vonzatkeretek is, amelyek hibátlanok, pusztán nem fordultak elő a Szeged Dependencia Treebankben, így kerültek ebbe a kategóriába (pl. *felvág vmivel*). A későbbiekben szeretnénk részletesebben is megvizsgálni, hogyan lehet automatikus eszközökkel tovább csökkenteni a hibás vonzatkeretek számát.

6 Összegzés

A cikkben bemutattuk a HunLearner korpuszt, mely a magyart mint idegen nyelvet tanulók által létrehozott szövegeket tartalmaz. A korpusz tartalmazza a morfológiailag hibás főnevek javított alakjait és a hiba kódját is. A javított alakok kézi annotációja lehetővé tette azt is, hogy megvizsgáljuk a hibák automatikus javításának lehetőségeit. Az eredmények azt mutatják, hogy már egyszerű módszerekkel is jelentősen lehet

csökkenteni a hibás szóalakok számát egy nem sztenderd szövegben, ami ígéretesnek mutatkozik a nem sztenderd szövegek automatikus feldolgozására nézve.

A jövőben tervezzük a korpusz további bővítését, továbbá szeretnénk feltérképezni a szintaktikai és szóhasználati hibák automatikus módszerekkel történő javításának lehetőségeit. A korpusz kutatási célokra szabadon elérhető a http://www.inf.u-szeged.hu/rgai/hunlearner oldalon.

Köszönetnyilvánítás

A kutatás a TÁMOP-4.2.2/C-11/1/KONV-2012-0013 jelű futurICT projekt keretében az Európai Unió és az Európai Szociális Alap társfinanszírozásával valósult meg. Vincze Veronikát az A/11/83421 jelű fiatal kutatói ösztöndíj keretében a Deutscher Akademischer Austauschdienst támogatta.

Hivatkozások

1. Centre for English Corpus Linguistics (UCL) [http://www.uclouvain.be/en-cecl-lcWorld.html]
2. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the Eighth International Conference on Text, Speech and Dialogue (TSD 2005). Karlovy Vary, Czech Republic 12-16 September, and LNAI series Vol. 3658 (2005) 123-131
3. De Cock, S., Granger, S.: Computer Learner Corpora and Monolingual Learners' Dictionaries: the Perfect Match. Lexicographica, Vol. 20 (2005) 72–86
4. Dickinson, M., Ledbetter, S.: Annotating Errors in a Hungarian Learner Corpus. In: Proceedings of the 8th Language Resources and Evaluation Conference (LREC 2012). Istanbul, Turkey (2012)
5. Durst P.: A magyar mint idegen nyelv elsajátításának vizsgálata – különös tekintettel a főnévi és igei szótövekre, valamint a határozott tárgyas ragozásra. Bölcsészdoktori értekezés. Kézirat. Pécs (2010)
6. Granger, S.: A Bird's-eye View of Computer Learner Corpus Research. In: Granger S., Hung J., Petch-Tyson, S. (eds): Computer Learner Corpora, Second Language Acquisition, and Foreign Language Teaching. Amsterdam & Philadelphia, Benjamins (2002) 3–33
7. Granger, S.: The computer learner corpus: A versatile new source of data for SLA research. In: Granger, S. (ed.): Learner English on Computer. London, Addison Wesley Longman Limited (1998) 3–18
8. Hana, J., Rosen, A., Škodová, S., Štindlová, B.: Error-Tagged Learner Corpus of Czech. In: Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010. (2010) 11−19
9. Jantunen, J. H.: Kansainvälinen oppijansuomen korpus (ICLFI): typologia,taustamuuttujat ja annotointi [International Corpus of Learner Finnish (ICLFI): typology, variables and annotation]. Lähivõrdlusi. Lähivertailuja Vol. 21 (2011) 86–105
10. Selinker, L.: Interlanguage. IRAL, Vol. 10 (1972) 209–230
11. Szirmai M.: Bevezetés a korpusznyelvészetbe. Budapest, Tinta Kiadó (2005)

12. Trón, V., Németh, L., Halácsy, P., Kornai, A., Gyepesi, Gy., Varga, D.: Hunmorph: open source word analysis. In: Proceedings of ACL (2005)
13. Vincze, V. Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (2010)
14. Vincze, V.: Valency frames in a Hungarian corpus. Kézirat (2012)
15. Zsibrita J., Vincze V., Farkas R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 368-374

Automatikus korpuszépítés tulajdonnév-felismerés céljára

Nemeskey Dávid Márk[1], Simon Eszter[2]

[1] MTA SZTAKI
1111 Budapest, Lágymányosi utca 11., e-mail:nemeskey.david@sztaki.mta.hu

[2] MTA Nyelvtudományi Intézet
1068 Budapest, Benczúr u. 33., e-mail: simon.eszter@nytud.mta.hu

Kivonat A felügyelt gépi tanulási módszerek alkalmazásához nagyméretű annotált korpuszokra van szükség, amelyek előállítása rendkívül emberierőforrás-igényes. Több lehetőség van az annotációs költségek csökkentésére, ezek közül az egyik az automatikus annotálás. Cikkünkben egy nyelvfüggetlen módszert mutatunk be, mellyel bármely Wikipédiával rendelkező nyelvre előállítható tulajdonnévi címkéket tartalmazó korpusz. Az automatikus annotálás során a DBpedia ontológiai kategóriáit képeztük le CoNLL-névosztályokra. Cikkünkben a magyar korpusz részletes hibaelemzését és kiértékelését adjuk.

Kulcsszavak: tulajdonnév-felismerés, korpuszépítés, automatikus annotáció, Wikipédia

1. Bevezetés

Az automatikus tulajdonnév-felismerés (Named Entity Recognition, NER) a természetes nyelv feldolgozását célzó alkalmazások közül az egyik legnépszerűbb, mivel hatékonyan automatizálható, és eredménye hasznos bemenete különböző magasabb szintű információkinyerő és -feldolgozó rendszereknek. A feladat során strukturálatlan szövegben kell azonosítani és az előre definiált osztályok valamelyikébe besorolni a neveket. A tulajdonnév-felismerés feladata a 6. Message Understanding Conference (MUC) egyik versenykiírásában jelent meg először 1995-ben [1]. Itt három alfeladatot különítettek el: tulajdonneveket, temporális és különböző numerikus kifejezéseket kellett felismerni. A NER-közösségen belül a temporális és a numerikus kifejezések annotálása is elfogadott, de a leginkább vizsgált típusok a személy-, földrajzi és intézménynevek. Ezek mellé vezettek be a CoNLL-versenyeken [2,3] egy negyedik típust, amely az előző háromba nem tartozó egyéb tulajdonneveket foglalja magában. Az azóta eltelt időben ezek az annotációs sémák váltak nemzetközileg elfogadottá.

A versenyekre épített és aztán közzétett tulajdonnév-annotált korpuszok képezik azokat a sztenderdeket, amelyek összemérhetővé teszik az egyes névfelismerő rendszereket. Ezek a korpuszok meglehetősen korlátozott méretűek és témaspecifikusak. Kellően robusztus tulajdonnév-felismerő rendszerek építéséhez

viszont nagyméretű, a téma tekintetében heterogén korpuszokra van szükség. A kézi annotálás rendkívül idő-, erőforrás- és szakértelemigényes feladat, ezért az elmúlt időkben különösen nagy hangsúly került az annotált erőforrások automatikus előállítására. Ennek egy módja, ha már rendelkezésre álló korpuszokat dolgozunk össze; ekkor a különböző annotációs sémák és címkekészletek összeillesztése állít elénk problémákat. Egy másik lehetőség az olyan webes közösségi tartalmak felhasználása korpuszépítéshez, mint például a Wikipédia, a Wiktionary vagy a DBpedia. Megint másik megközelítés az annotáció automatizálása, ami az esetek nagy részében egy már rendelkezésre álló adathalmazon tanított rendszer új szövegen való futtatását jelenti.

Cikkünkben egy olyan megközelítést mutatunk be, mely ezen lehetőségeket kombinálja: automatikus eszközökkel tulajdonnév-annotált korpuszokat építettünk Wikipédia szócikkekből. Munkánk során új módszert alkalmaztunk: a DBpedia ontológiai kategóriáit képeztük le CoNLL-névosztályokra. A módszert egyelőre a magyar és az angol Wikipédiára alkalmaztuk.

A cikk a következőképpen épül fel. A 2. fejezetben bemutatjuk a Wikipédia eddigi felhasználási módjait a tulajdonnév-felismerés területén. A 3. fejezetben leírjuk a korpuszépítési módszert, elsősorban a magyar nyelvű adatokra koncentrálva. Az alkalmazott módszer részletes hibaelemzését a 4., a korpuszok leírását a 5., míg a kiértékelést és az eredményeket a 6. fejezet adja. Cikkünket az elért eredmények rövid összefoglalása zárja (7. fejezet).

2. Wikipédia és tulajdonnév-felismerés

A Wikipédia egy többnyelvű, nyílt tartalmú, az internetes közösség által fejlesztett webes világenciklopédia[3]. Több mint 22 millió szócikkével kincsesbánya a különböző természetesnyelv-feldolgozó fejlesztések számára; használták már többek között jelentésegyértelműsítére, ontológia- és tezauruszépítésre, valamint kérdésmegválaszoló rendszerekhez (további alkalmazási lehetőségekért lásd [4]). Mivel a Wikipédia címszavainak jelentős része tulajdonnév, adja magát a lehetőség, hogy a tulajdonnév-felismeréshez is használjuk.

A Wikipédia legkézenfekvőbb alkalmazási módja nagyméretű névlisták előállítása, melyek javítják az általános célú névfelismerők hatékonyságát, felügyelt és felügyelet nélküli módszerek esetében is (pl. [5] és [6]), továbbá a név-egyértelműsítésben is fontos szerepet játszanak (pl. [7]). A Wikipédiában található tudás jegyek formájában is beépíthető tulajdonnév-felismerő rendszerekbe: például Kazama és Torisawa [8] kísérlete azt bizonyítja, hogy a Wikipédia kategóriacímkéinek automatikus kinyerése növeli egy felügyelt névfelismerő rendszer pontosságát.

A Wikipédia alkalmazására a névfelismerés területén egy másik lehetőség magának a szövegbázisnak a felhasználása. Richman és Schone [9] kevés erőforrással rendelkező nyelvek Wikipédia-cikkeiből épített korpuszokat, amelyekben a Wikipédia inherens kategóriastruktúráját használták fel a tulajdonnevek annotálásához. Nothman et al. [10] a szócikkek első mondatából kiindulva címkézte fel

[3] http://wikipedia.org

a szövegben belinkelt neveket, így építve automatikusan tulajdonnév-annotált korpuszt.

Az általunk alkalmazott módszer az említettektől annyiban tér el, hogy mi a DBpedia ontológiai osztályait képeztük le a sztenderd CoNLL-névosztályokra, majd ezeket Wikipédia-entitásokhoz kötöttük. A Wikipédiából, tekintve a szócikkek nagy számát, kevés erőforrással rendelkező nyelvekre is tudunk kellően nagy méretű korpuszokat építeni, amelyek bemenetül szolgálhatnak névfelismerő rendszerek tanításához és teszteléséhez. Tudomásunk szerint az általunk létrehozott korpusz az első magyar nyelvű automatikusan tulajdonnév-annotált korpusz, amely szabadon elérhető és felhasználható további kutatásokhoz.

3. Korpuszépítés

A korpuszépítő algoritmus a Wikipédia két feltételezett tulajdonságán alapszik, miszerint a szócikkek többsége megnevezett entitásokról szól, valamint a folyó szövegben előforduló entitásokat a kereszthivatkozások azonosítják. Algoritmusunk e két feltételezés következményeit használja ki. Hasonlóan a Nothman et al. [10] által leírtakhoz, a korpuszépítés a következő lépésekből áll:

1. a Wikipédia-cikkeket entitásosztályokba soroljuk;
2. a cikkeket mondatokra bontjuk;
3. felcímkézzük a tulajdonneveket a szövegben;
4. kiszűrjük a rossz minőségű mondatokat.

Az algoritmus alapjaiban nyelvfüggetlen: bármely nyelvre alkalmazható, amely rendelkezik megfelelő méretű Wikipédiával. Egyedül a harmadik lépés az, ahol figyelembe kell venni a nyelv, illetve a használt annotációs séma sajátosságait. Ennek oka, hogy az egyes nyelvek, illetve sémák eltérnek abban, hogy mit tekintenek annotálandó elemnek: pl. a *római* szó a magyarban nem számít névnek, míg angol megfelelője, a *Roman* a CoNLL-séma szerint `Misc` címkét kapna.

Mivel célunk egy minél tisztább, vagyis a gold standard színvonalat közelítő korpusz előállítása volt, ha választanunk kellett egy-egy lépésnél a pontosság (precision) vagy a teljesség (recall) között, mindig az előbbi mellett döntöttünk. A Wikipédia mérete lehetővé teszi, hogy szigorú szűrések mellett is korábban nem látott méretű korpuszt állítsunk elő.

A továbbiakban röviden ismertetjük a fenti lépéseket, kizárólag a magyar nyelvű korpuszra koncentrálva. Részletes leírás, illetve az angol nyelvű korpuszban felmerülő problémák kifejtése [11]-ben található.

3.1. Wikipédia-cikkek mint entitások

Ahhoz, hogy a folyó szövegben lévő kereszthivatkozásokat felhasználhassuk a tulajdonnevek azonosítására, a hivatkozott Wikipédia-cikkeket névosztályokba kell sorolnunk. A Wikipédia saját kategóriarendszere a kategóriák nagy száma, illetve

a rendszerezettség teljes hiánya miatt nem alkalmas erre a célra. Egyes szerzők, mint Kazama és Torisawa [8] vagy Nothman et al. [10] felügyelt tanulással oldották meg ezt a feladatot. Mi azonban a klasszifikációs hibák elkerülése céljából a DBpedia [12] típushierarchiájának felhasználása mellett döntöttünk.

A DBpediában[4] a típusok egy OWL[5] ontológia részei. A tudásbázis a Wikipédia-entitások egy részhalmazát tartalmazza, és mindegyik entitáshoz hozzárendeli – többet között – azon ontológiaosztályokat, amelyekbe az tartozik. Mivel a DBpediának nincs magyar változata, a magyar entitáslista csak olyan angol oldalak adatait tartalmazza, melyeknek létezik megfelelője a magyar Wikipédiában. Ezáltal a feldolgozás köréből kiesnek kifejezetten magyarspecifikus oldalak, de így is 58.337 oldal anyagát dolgoztuk fel.

A magyar nevek kategorizálásához a Szeged NER [13] korpuszban alkalmazott névosztályokat vettük alapul, ezekre képeztük le a DBpedia egyes ontológiaosztályait. Ezután minden entitáshoz hozzárendeltük az őt tartalmazó legszűkebb osztály címkéjét, vagy `O`-t, ha az osztály nem minősül annotálandónak a Szeged NER korpusz sémája szerint. Így egy 46.461 elemű, felcímkézett, angol nyelvű tulajdonnévlistát kaptunk. Utolsó lépésként meghatároztuk ezen oldalak magyar és egyéb nyelvű megfelelőit, és ezeket is felvettük a listára. Az idegen nyelvű oldalak meghagyásának oka, hogy a magyar Wikipédia időnként ezekre is hivatkozik.

3.2. A cikkek feldolgozása

A korpusz építéséhez a magyar Wikipédia 2012. március 9-i állapotát vettük alapul. Az XML-fájlokból az mwlib[6] könyvtár segítségével nyertük ki a nyers szöveget. A tokenizáláshoz egy házon belül előállított statisztikai eszközt használtunk, amelyet a Szeged korpuszon [14] tanítottunk. A lemmatizálást és a morfológiai elemzést a HunMorph [15] és az erre épülő egyértelműsítő, a HunDisambig segítségével végeztük.

3.3. Tulajdonnevek címkézése

A címkézés két feladatot foglal magába: egyrészt azonosítani kell az entitásokat a folyó szövegben, másrészt besorolni őket a megfelelő névosztályba. A szövegben előforduló kereszthivatkozásokat potenciális neveknek tekintjük, és minden linknél megnézzük, hogy az oldal, amire mutat, szerepel-e a korábban előállított listában (lásd a 3.1. fejezetben). Ha igen, felcímkézzük a megfelelő címkével; ha nem, típusa `Unk` lesz. Végül minden mondatot, amelyben `Unk`-ként címkézett elem szerepel, eldobunk.

Kiinduló alapfeltételezésünk, miszerint az egyes Wikipédia-cikkek megnevezett entitásokról szólnak, és a szövegben előforduló entitásokat kereszthivatkozások azonosítják, nem minden esetben állja meg a helyét. Algoritmusunk ezért több helyen finomításra szorul.

[4] A korpusz elkészítéséhez az akkor aktuális 3.7-es verziót használtuk.
[5] http://www.w3.org/TR/owl-ref/
[6] http://code.pediapress.com

Először, a Wikipédia nem minden szócikke szól megnevezett entitásokról: egyes köznevek, dátumok és egyéb, nem tulajdonnévi elemek is kaptak saját oldalt. Annak érdekében, hogy az algoritmus ne kezelje ezeket `Unk` típusú entitásként, majd dobja el a mondatokat, amikben szerepelnek, szigorítottuk az entitásfelismerés szabályait: csak olyan kereszthivatkozást tekintünk potenciális névre való utalásnak, melynek szavai nagybetűvel kezdődnek. Itt és később is kihasználtuk a magyar nyelv azon tulajdonságát, hogy a tulajdonneveket, és csak azokat kezdjük nagybetűvel. Kivételt képeznek ez alól természetesen a mondatkezdő pozícióban szereplő szavak. Szigorúan véve, ez a módszer minden mondatot eldobna, így annyit módosítottunk rajta, hogy a mondatkezdő szót csak akkor tekintjük potenciális annotálandó elemnek, ha a szófaja főnév. (A magyarban a morfológiai címkékre csak korlátozottan támaszkodhattunk ebben a feladatban, hiszen a KR-kódolás nem különbözteti meg a tulajdonneveket és a közneveket.)

Másodszor, nem minden tulajdonnéven találunk kereszthivatkozást. Ennek oka kettős: ha egy Wikipédia-cikkben adott entitásra többször is névvel utalnak, csak az első alkalommal linkelik a saját oldalához. De előfordulhat az is, hogy az entitás nem rendelkezik saját szócikkel. Az ilyen esetek kezelésére minden szócikknél fenntartunk egy listát, ahol az abban felismert entitásokat, illetve egyéb nagybetűs említéseit, mint például a rájuk mutató átirányító és egyértelműsítő lapok címeit gyűjtjük. Ha ezután egy olyan mondattal találkozunk, amelyben nagybetűs szócsoport szerepel, ellenőrizzük, hogy a csoport egyezike a listában szereplő egyik entitás nevével. Amennyiben igen, felcímkézzük; ha nem, a nagybetűs szavakat ismeretlen entitásnak tekintjük.

3.4. Szűrés

Ahogy már említettük, az azonosítatlan entitást tartalmazó mondatokat kidobtuk a korpuszból, hogy növeljük az annotálás pontosságát.

A hagyományos névfelismerő rendszerek tiszta, viszonylag pontosan annotált korpuszokból tanulják meg a megfelelő paramétereiket, és a tesztelésükhöz is hasonló adathalmazra van szükség. Ezért a rossz minőségű, töredékes mondatokat, amelyek nem nagybetűvel kezdődnek és nincs mondatzáró írásjel a végükön, szintén kiszűrtük. (Az így maradt 19 milliós adathalmazzal dolgoztunk tovább; a következőkben erre referálunk teljes korpuszként.) De a minőség javítása érdekében további szűrő lépéseket is lehet tenni, így például eldobhatjuk az olyan mondatokat is, amelyek nem tartalmaznak ragozott igét. Ezzel ugyan azok is kiesnek, amelyek szabályos jelen idejű, kijelentő módú, létigét (nem) tartalmazó mondatok, de az automatikus tokenizálás és mondatra bontás hibáiból származó töredékes mondatokat eltávolíthatjuk, és így az ebből fakadó annotációs hibákat is kiszűrhetjük (lásd a 4. fejezetet).

Ha viszont közösségi tartalmak (User Generated Content) feldolgozására kívánja valaki használni a korpuszokat, amelyek köztudottan sokkal zajosabbak, mint a kézzel annotált adatok, és sok töredékes mondatot tartalmaznak, hasznos lehet minél kevesebb szűrő használata. Ezért a rossznak minősített mondatokat nem dobtuk el végleg, hanem ezt az adathalmazt is elérhetővé tettük.

4. Problémás esetek, hibaelemzés

A magyar Wikipédia korpusz gépi annotálását kézzel ellenőriztük a korpusz egy kis részén. A 19 milliós teljes korpuszt vettük alapul, amelyből véletlenszerű mondatválogatással csináltunk egy 18.830 tokent tartalmazó mintakorpuszt. Ezt kézzel felannotáltuk, és összehasonlítottuk a kézi és a gépi annotálás eredményét, amely az 1. táblázatban látható. Ha a gépi módszert egy annotátornak tekintjük, akkor az F-mérték az annotátorok közötti egyetértést mutatja.

1. táblázat. A gépi és a kézi annotálás közötti egyetértést mutató eredmények a mintakorpuszon.

	Pontosság(%)	Fedés(%)	$F_{\beta=1}$(%)	Entitások száma
LOC	98.72	95.65	97.16	161
MISC	95.24	76.92	85.11	26
ORG	89.66	89.66	89.66	29
PER	88.30	89.25	88.77	93
Összesítve	94.33%	91.59%	92.94	309

A négy kategória igazságmátrixa (2. táblázat) jól mutatja, hogy a típustévesztés aránya elhanyagolható. Ezekből az értékekből kiszámítottuk az annotátorok közötti egyetértést mérő Cohen kappát is. A 0,967-es összesített érték Landis és Koch [16] skáláján elhelyezve arról tanúskodik, hogy a korpusz annotációja megfelel a gold standard színvonalnak.

2. táblázat. A manuálisan annotált mintakorpusz igazságmátrixa.

Auto↓ / Gold→	PER	ORG	LOC	MISC
PER	83	1		2
ORG		26	1	1
LOC		1	154	
MISC			1	20

A típustévesztés alapvetően két okra vezethető vissza. Az első esetben a DBpediában lévő típusinformáció helytelen, például az *MTA* DBpedia-beli osztálya `WorldHeritageSite`, ami miatt `Loc` címkét kap `Org` helyett. Hasonló eset, amikor egy több referenciával rendelkező névnek csak egyik referense szerepel a DBpediában, így használattól függetlenül mindig ugyanazt a címkét kapja.

A típustévesztések másik részét a Wikipédia rossz kereszthivatkozásai okozzák. Például az egyik cikk szerkesztője a *Walt Disney Co.* cégnévnek csak egy

részét linkelte be, mégpedig a személyről szóló oldalra. Ezért ebben a cikkben ennek a cégnévnek a különböző változatai (*Disney*, *Walt Disney* stb.) is mind személynévként lettek jelölve.

3. táblázat. A névfelismerés további hibái.

	PER	ORG	LOC	MISC
Hibásan névként felismert szavak (álpozitív)	1	0	1	0
Fel nem ismert nevek (álnegatív)	3	0	5	4
Részlegesen felismert nevek	7	1	0	0

Jóval gyakoribbak a névfelismerés további hibái, vagyis bizonyos szavak hibásan névként való azonosítása és egyes nevek felismerésének elmulasztása, illetve az entitáshatárok pontatlan felismerése. Az egyes névtípusokra lebontott hibák számát a 3. táblázat mutatja.

Az entitáshatárok pontatlan felismerésének oka leggyakrabban az, hogy a név környezetében lévő értelmező szerepű nyelvi egység is a Wikipédia-címszó része. Emberekre utaló linkek esetében ezek a szavak nagyrészt rangjelölők, pl. *Szent István király*, *I. Benedek pápa*. Ezeket egy tematikus tiltólistával a későbbiekben ki lehet szűrni.

A Wikipédia-címszavakban szereplő értelmező szavak alkalmanként a teljes entitás felismerését is megakadályozzák. Ez akkor fordul elő, ha a hivatkozott oldal teljes címe nem tulajdonnév, viszont tartalmaz egy általunk ismert tulajdonnevet: pl. *ókori Róma*, *magyar Wikipédia*.

A fel nem ismert `Misc` típusú tulajdonnevek mindegyike írók műveit felsoroló oldalakon fordul elő. Ezen műveknek nincs saját szócikkük, ezért címkézésük nehéz. Mivel egy műcímben bármilyen nyelvi elem előfordulhat, az általunk alkalmazott szűrők együttese sem képes kiszűrni ezeket. Megoldást jelenthetne az, ha ezeket a kizárólag műcímeket felsoroló oldalakat külön kezelnénk, és egy komplex rendszert építenénk a feldolgozásukra. Mivel ez egy külön nyelvfeldolgozási feladat, jelen fejlesztésen belül nem vállalkozunk rá; a jövőben az ilyen oldalakat kihagyjuk a korpuszból.

A hibák egy további részét az alkalmazott nyelvfeldolgozó eszközök tévesztései okozzák. Az automatikus tokenizálás és mondatra bontás hibás működésére példa, amikor a rövidítést tartalmazó név (pl. *Warner Bros.*) utolsó eleme, vagyis a pont mondatvégi írásjelként értelmeződik, így nem annotálódik a névvel együtt. Máskor a mondatrabontás során a szövegben levő link szétszakad, így a maradék elem nem kapja meg a megfelelő címkét. Mivel a mondat első szavát csak akkor tekintjük potenciális entitásnak, ha főnév, a szófajmeghatározás hibája folytán előfordul, hogy átsiklunk egy mondatkezdő néven. Például a *Hél visszaengedte volna* mondatban a *Hél* szót igeként azonosította a HunDisambig, így nem tekintettük tulajdonnév-jelöltnek. E problémákra esetleg megoldást jelenthet az alkalmazott eszközök teljesítményének javítása, vagy más eszközök használata.

Tipikus jelenség a magyarban, amikor egy név összetétel eleme lesz, pl. *Bizánc-ellenes*. Ilyen esetekben a köznév az összetétel alaptagja, vagyis az határozza meg a referenciát. A referenciaváltozást természetesen a címkézés változásának kell követnie, ami viszont nem, vagy nehezen kezelhető automatikusan, mivel nagyjából bármilyen köznév kapcsolódhat névhez. A helyzetet bonyolítja az is, hogy a mozaikszavakhoz, rövidítésekhez, valamint nem ejtett magánhangzóra végződő idegen nevekhez is kötőjellel kapcsoljuk a toldalékokat, amelynek a felszíni szerkezete nagyon hasonlít az összetételekéhez. Ennek a problémának a megoldása még további vizsgálatokat követel.

A felsoroltak mellett természetesen implementációs hibákra is fény derült, amelyek azonban összességében csak néhány esetben okoztak rossz címkézést. Ezeket a jövőben javítani fogjuk.

5. A korpuszok leírása

A korpuszokat Creative Commons Attribution-Sharealike 3.0 licensz alatt publikáljuk, vagyis ugyanolyan feltételekkel adjuk tovább, ahogy a Wikipédiából letöltöttük. Szabadon elérhetőek a `http://hlt.sztaki.hu` oldalon keresztül, valamint a META-SHARE tárhelyről (http://www.meta-net.eu/). A META-SHARE egy nyílt rendszer, amely lehetővé teszi a nyelvi erőforrások megosztását. Létrehozója a META-NET, az Európai Bizottság által alapított nyelvtechnológiai hálózat.

A fájlok ún. *multitag* formátumban vannak, amelyben a tartalmas sorokat tabulátor választja el. Az első oszlop tartalmazza magukat a szövegszavakat, az egyes oszlopokban pedig a különböző szintű annotációk találhatók. A mondathatárokat üres sorok jelölik. A névcímkéken kívül minden token mellett szerepel a töve és a hozzá tartozó teljes morfológiai elemzése KR-kódokkal. Két további oszlopban közöljük, hogy sima szöveg vagy kereszthivatkozás-e az adott token, és ha utóbbi, akkor melyik szócikkre utal.

6. Kiértékelés

Kiértékelésünkben megmutatjuk, hogy a létrehozott magyar nyelvű korpusz kiválóan használható a tulajdonnév-felismerés teljesítményének növelésére több módon is.

A kiértékeléshez a Hunner [17] tulajdonnév-felismerő rendszert használtuk. A csak az egyes korpuszokra jellemző jegyeket (pl. főnévi csoportok jelölése, Wikipédia-linkek) kidobtuk, hogy növeljük a korpuszok összehasonlíthatóságát. Így a következő jegykészlettel dolgoztunk: mondatkezdő és -vég pozíciók, szóalakon alapuló jegyek, morfológiai információ és listajegyek.

Az eredmények kiszámításához a sztenderd CoNLL-módszert alkalmaztuk, vagyis az annotációt csak akkor vettük helyesnek, ha a kezdő- és végpozíció is stimmelt, és a rendszer által kibocsátott címke megegyezett a gold standard címkével. Ezen alapulva a szokásos pontosságot, fedést és F-mértéket számoltuk.

6.1. Az adatok

A korpusz a fent leírt szűrő eljárások után maradt mondatokat tartalmazza, így azokat is, amelyekben nincs egy név sem. Ezeket azért tartottuk meg, hogy amennyire lehetséges, megőrizzük a nevek eredeti, Wikipédia-beli eloszlását. Viszont amikor megvizsgáltuk az egyes korpuszok telítettségét a nevek szempontjából, arra jutottunk, hogy a gold standard adathalmazzal való összevetéskor inkább sűrítjük a szöveget, vagyis kivesszük azokat a mondatokat, amelyekben nincs név. A 4. táblázat mutatja a magyar korpuszokra vonatkozó számszerű adatokat, melyekből jól látható, hogy a Wikipédiából generált korpusz telítettsége meglehetősen alacsony. A szövegnek ez a hígsága valószínűleg annak köszönhető, hogy a módszerünk szigorú, vagyis inkább minden olyan mondatot eltávolítottunk, amelyben nem lehetett beazonosítani a nevet, minthogy rosszul annotált nevek maradjanak benne.

4. táblázat. A magyar Wikipédia és a Szeged NER korpusz mérete és telítettsége.

	huwiki	**sűrített huwiki**	**Szeged NER**
token	19.108.027	3.512.249	225.963
NE	456.281	456.281	25.896
telítettség (%)	2,38	12,99	11,46

6.2. Kísérletek és eredmények

Jelen cikkben csak a magyar korpuszon elért eredményeket közöljük, az angolra vonatkozó részletes adatokért lásd korábbi cikkünket [11]. A korpusz kétféleképpen lett kiértékelve: először saját magán, aztán egy választott gold standard adathalmazon. A nevet nem tartalmazó mondatok kiszűrése után maradt 3,5 millió tokenes korpuszt 90-10%-os arányban tanító és kiértékelő halmazra osztottuk.

Mivel a névkategóriák leképezésénél a Szeged NER korpusz címkekészletét használtuk, ezért adta magát, hogy a korpusz kiértékeléséhez is ugyanezt alkalmazzuk. Többek által (pl. [10] és [18]) bizonyított tény, hogy a korpuszok közötti kiértékelés sokkal rosszabb eredményt ad, mint a saját kiértékelő halmazon való mérés. Különböző típusú szövegek esetén a különbség 20-30% is lehet. A helyzet a mi esetünkben is nagyon hasonló (lásd az 5. táblázatot az eredményekért): a Wikipédián tanított rendszer teljesítménye közel sem olyan jó a gold standard korpusz kiértékelő halmazán mérve, mint a saját kiértékelő halmazán.

Az általunk épített korpuszt további módokon is használhatjuk a tulajdonnév-felismerés teljesítményének növelése érdekében. Egy kézenfekvő megoldás nagyméretű névlisták kinyerése a Wikipédiából, és azok hozzáadása gazetteer listaként a tanításhoz. Ez a módszer több mint 1%-kal növelte az F-mértéket.

5. táblázat. Eredmények a magyar Wikipédia korpuszon.

tanítás	teszt	Pontosság (%)	Fedés (%)	F-mérték (%)
Szeged	Szeged	94,50	94,35	94,43
huwiki	huwiki	90,64	88,91	89,76
huwiki	Szeged	63,08	70,46	66,57
Szeged_wikilisták	Szeged	95,48	95,48	95,48
Szeged_wikitag	Szeged	95,38	94,92	95,15

Egy másik kísérletünkben a rendszert a Wikipédia korpuszon tanítottuk, majd az általa kibocsátott címkéket jegyként hozzáadtuk a gold standard korpuszon való tanításhoz és teszteléshez. Ezzel a módszerrel is sikerült javítani a rendszer teljesítményét.

A kiértékelés legfontosabb eredményének a saját teszthalmazon elért 89,76%-os F-mértéket tartjuk. A kézi hibaelemzés tanulságaival együtt ez arról tanúskodik, hogy az általunk épített korpusz akár önálló gold standard adathalmazként, akár kiegészítő erőforrásként jól használható automatikus névfelismerő rendszerek építéséhez.

7. Összegzés

Cikkünkben egy új módszert mutattunk be, amellyel létrehoztunk egy magyar nyelvű, automatikusan tulajdonnév-annotált korpuszt a Wikipédiából. Az eddig alkalmazottakkal ellentétben a mi metódusunk egy leképezést valósít meg a DBpedia ontológiai osztályairól a hagyományos címkekészletekre. Az így generált címkéket aztán a rendszer hozzárendeli a Wikipédiában szereplő entitásokhoz.

Módszerünk nyilvánvaló előnyei, hogy nagyban csökkenti az annotálás költségeit, valamint hogy sokkal nagyobb adathalmazokat állíthatunk elő általa, mint kézi annotációval. Egy másik előnye, hogy bármely Wikipédiával rendelkező nyelvre alkalmazható, így kevés erőforrással rendelkező nyelvekre is előállíthatunk a gold standard minőséget közelítő korpuszokat. A létrehozott korpuszok a továbbiakban számos módon alkalmazhatók a tulajdonnév-felismerő rendszerek hatékonyságának növelésére. Amennyiben kellően tiszta a korpusz, vagy az adott nyelvre nem létezik gold standard tisztaságú adathalmaz, felügyelt gépi tanulási rendszerekhez használható tanításhoz és kiértékeléshez. Továbbá erőforrásokkal bővebben ellátott nyelvek esetében is hasznosítható a klasszikus sajtó stílustól eltérő szövegek tulajdonnév-annotálásához.

További, újdonságnak számító eredményünk, hogy az általunk előállított korpuszok szabadon elérhetőek és felhasználhatóak. Tudomásunk szerint ez az első magyar nyelvű automatikusan előállított tulajdonnév-annotált korpusz. Az angol erőforrások tekintetében is hasonló a helyzet: tudomásunk szerint a Semantically Annotated Snapshot of English Wikipedia [19] mellett az itt publikált korpusz az egyetlen szabadon felhasználható tulajdonnév-annotált korpusz.

Jelen cikkünkben a DBpedia ontológiai kategóriáit a sztenderd tulajdonnév-címkékre képeztük le, de a módszerben benne rejlik a lehetőség finomabbra hangolt tulajdonnév-hierarchiák támogatására is. Az internetes közösség által létrehozott tartalmak, mint a Wikipédia és a DBpedia, folyamatosan növekszenek, ezáltal egyre több információ felhasználását teszik lehetővé. A módszer frissítésével egyre nagyobb és finomabban annotált korpuszokat tudunk létrehozni a jövőben.

Köszönetnyilvánítás

A fejlesztés az OTKA 82333. számú projektjén belül valósult meg. A fejlesztést támogatta továbbá a CESAR projekt (No. 271022). A szerzők ezúton fejezik ki köszönetüket Zséder Attilának a Wikipédia-szövegek feldolgozásában végzett munkájáért, és Kornai Andrásnak támogatásáért.

Hivatkozások

1. Sundheim, B.: MUC-6 Named Entity Task Definition (v2.1). In: Proceedings of the Sixth Message Understanding Conference (MUC6). (1995)
2. Sang, T.K., F., E.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei, Taiwan (2002) 155–158
3. Sang, T.K., F., E., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2003, Edmonton, Canada (2003)
4. Medelyan, O., Milne, D., Legg, C., Witten, I.H.: Mining meaning from Wikipedia. Int. J. Hum.-Comput. Stud. **67**(9) (2009) 716–754
5. Toral, A., Munoz, R.: A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In: EACL 2006. (2006)
6. Nadeau, D., Turney, P., Matwin, S.: Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. Advances in Artificial Intelligence (2006) 266–277
7. Bunescu, R., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. (2006) 9–16
8. Kazama, J., Torisawa, K.: Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. (2007) 698–707
9. Richman, A.E., Schone, P.: Mining Wiki Resources for Multilingual Named Entity Recognition. In: Proceedings of ACL-08: HLT, Columbus, Ohio, Association for Computational Linguistics (2008) 1–9
10. Nothman, J., Curran, J.R., Murphy, T.: Transforming Wikipedia into named entity training data. In: In Proceedings of the Australasian Language Technology Association Workshop 2008. (2008) 124–132
11. Simon, E., Nemeskey, D.M.: Automatically generated NE tagged corpora for English and Hungarian. In: Proceedings of the 4th Named Entity Workshop (NEWS) 2012, Jeju, Korea, Association for Computational Linguistics (2012) 38–46

12. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – A crystallization point for the Web of Data. Web Semantics **7**(3) (2009) 154–165
13. Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.: A highly accurate Named Entity corpus for Hungarian. In: Electronic Proceedings of the 5th International Conference on Language Resources and Evaluation. (2006)
14. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus. A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus. In Hansen-Schirra, S., Oepen, S., Uszkoreit, H., eds.: COLING 2004 5th International Workshop on Linguistically Interpreted Corpora, Geneva, Switzerland, COLING (2004) 19–22
15. Trón, V., Gyepesi, Gy., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: open source word analysis. In: Proceedings of the ACL 2005 Workshop on Software. (2005)
16. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics **33**(1) (1977) 159–174
17. Varga, D., Simon, E.: Hungarian named entity recognition with a maximum entropy approach. Acta Cybernetica **18** (2007) 293–301
18. Ciaramita, M., Altun, Y.: Named-entity recognition in novel domains with external lexical knowledge. In: Proceedings of the NIPS Workshop on Advances in Structured Learning for Text and Speech Processing. (2005)
19. Atserias, J., Zaragoza, H., Ciaramita, M., Attardi, G.: Semantically Annotated Snapshot of the English Wikipedia. In: Proceedings of LREC 2008. (2008)

IV. Pszichológia

Szemantikus szerepek a narratív kategoriális elemzés (NARRCAT) rendszerében

Ehmann Bea[1], Lendvai Piroska[2], Miháltz Márton[2],
Vincze Orsolya[3], László János[1,3]

[1] MTA TTK Kognitív Idegtudományi és Pszichológiai Kutatóintézet
1132 Budapest, Victor Hugó u. 18-22.
ehmannb@mtapi.hu
[2] Nyelvtudományi Intézet,
1068 Budapest, Benczúr u. 33.
piroska@nytud.hu; mmihaltz@gmail.com
[3] Pécsi Tudományegyetem, Pszichológiai Intézet
7624 Pécs, Ifjúság útja 6.
orsolyavincze@hotmail.com; laszlo@mtapi.hu

Jelen munkánk a digitális bölcsészet keretébe tartozik, s ezen belül korpusznyelvészek és narratív pszichológusok korábban már megkezdett együttműködésének fejleményeiről számol be [8,1,2,5]. A tudományos narratív pszichológia [4], s ezen belül a narratív szociálpszichológia [3] – a hagyományos pszichológiai tartalomelemzési megközelítéseken túlmenően – nem csupán strukturális vagy mintázatelemzést végez a szövegeken, hanem azt is vizsgálja, hogy az adott érzelem, kogníció, értékelés vagy cselekvés milyen cselekvőhöz, illetve milyen elszenvedőhöz tartozik. Ehhez szükséges a szemantikus szerepek (Semantic Role Labeling) vizsgálatára szolgáló elemzőeszköz kifejlesztése.

A Narratív Pszichológiai Munkacsoport – a korábbi pszichoszemantikai modulok alapján – kidolgozta a narratív kategoriális elemzés egységes módszertani eszköztárát, a NARRCAT-ot. E rendszer központi elemei a narratív kategoriális avagy pszichotematikus elemzőmodulok (Ágencia, Érzelem, Értékelés, Kogníció, Időbeliség és Térbeliség). A rendszer újdonsága, hogy külön modulként tételeződik a szemantikus szerepek vizsgálata (SRL), amely ezek közül az első néggyel kapcsolódik össze. Az SRL modul a Társas Referenciák nevű, ugyancsak önálló modulból kap inputot; ez mindenekelőtt azt határozza meg, hogy személyközi vagy csoportközi viszonyokról van szó; a csoportközi viszonyok vizsgálatában a rendszer képes beazonosítani a Saját Csoport (Ingroup) és Külső Csoport (Outgroup) ágenseket. A szemantikus szerep szerint az Ingroup és az Outgroup egyaránt lehet a cselekvés, az érzelem, az értékelés vagy a kogníció Ágense vagy Recipiense.

Korábbi törekvéseink során általános elveket (a MetaMorpho nyelvi elemzés morfoszintaktikai és szemantikai kimenetének összekapcsolása a NooJ eszköz procedúráival) és nagy korpuszokat (a TTK KPI archívumában található teljes történeti szövegkorpuszt) vontunk be a fejlesztésbe. Korábbi kiindulópontunkat a MetaMorpho által meghatározott igei NP-bővítmények képezték, s ehhez rendeltük hozzá a NooJ eszköz általunk készített pszichoszemantikus csoport szótárja alapján az InGroup, OutGroup, vagy MixGroup címkét [3]. Ezután a munkát két szálon folytattuk tovább: a korpusznyelvészeti fókuszú és a pszichológiai fókuszú kiindulópontból.

A korpusznyelvészeti szálhoz a pszichológusok hozzájárulása a névelemek kézi kilistázása és Ingroup/Outgroup kategóriába sorolása volt. (Ehhez a feladathoz a TTK KPI archívumában lévő történelmi korpuszok közül a 210000 szót tartalmazó általános és középiskolai tankönyvkorpuszt használtuk.) A korpusznyelvészek számára a szövegek előfeldolgozása az alábbi négy fő kihívást foglalja magába: a névelemek lemmatizálása, a koreferenciák feloldása, a metonímiák kezelése és az anaforafeloldás [6]. E szegmens megoldása után következik a tulajdonképpeni nyelvészeti SRL feladat. Ennek első fázisa az Agent és a Patient/Recipiens/Undergoer szerepek kézi annotálása, majd a további annotáció gépi támogatása. Ennek egyik területe a magyar WordNet lexiko-szemantikai adatbázisban [7] hasonlósági metrika alapján a hasonlónak ítélt V-k gépi előannotálása (Agent-Patient), szabályalapúan segítve az Ingroup/Outgroup szótárakkal. További cél egy Ingroup/Outgroup címkéző modell létrehozása: a szótárban eddig nem szereplő, Agent vagy Patient szemantikus szerepben lévő NPk automatikus címkézése Ingroup/ Outgroupként.

Az előadásban a pszichológusok által végzett munka új empirikus eredményeiről is be kívánunk számolni. Ennek lényege, hogy a Történelemtankönyv Korpuszból kézi annotálással kilistázott mintegy 9000 Ingroup/Outgroup névelem alapján létrehoztuk a NARRCAT rendszer történelemszöveg-specifikus SLR modulját, és ezt a Pszicho-tematikus modulokkal összekapcsolva, statisztikailag igazolt következtetéseket állapítottunk meg arra vonatkozóan, hogy az iskolai történelemtankönyvek hogyan reprezentálják a csoportidentitás komponenseit a különböző történelmi korokban.

A fentiekből két dolog világosan látható. Az egyik, hogy az SRL probléma korpusznyelvészeti megoldása egy többéves, nagy ívű projekt, melynek során akár egyes részeredmények is nagy előrelépést jelentenek. A másik, hogy a nyelvészek és a pszichológusok együttműködése, együttgondolkodása e téren kölcsönösen gyümölcsöző: a pszichológusok adják a korpuszt és az Ingroup/Outgroup névelemlistákat, másrészt beépítik empirikus munkájukba a nyelvészektől kapott tudást és eredményeket.

Bibliográfia

1. Ehmann B., Lendvai P., Fritz A., Miháltz M., Tihanyi L.: Szemantikus szerepek vizsgálata magyar nyelvű szövegek narratív pszichológiai elemzésében. In: Tanács A., Vincze V. (szerk.): VIII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2011) 223–230
2. Ehmann, B., Lendvai, P., Pólya, T., Vincze, O., Miháltz, M., Tihanyi, L., Váradi, T., László, J.: Narrative Psychological Application of Semantic Role Labeling. In: Vučković, K., Bekavac, B., Silberztein, M. (eds.): Automatic Processing of Various Levels of Linguistic Phenomena: Selected Papers from the NooJ 2011 International Conference, Cambridge Scholars Publishing, Newcastle upon Tyne, UK (2011) 218–228
3. László, J., Ehmann, B.: Narrative Social Psychology. In: Forgas, J., Vincze, O., László, J. (eds.): Social Cognition and Communication. Sydney Symposium in Social Psychology Series (Series Editor: Forgas, J.P.). Psychology Press, New York (2012) (In Press)
4. László, J.: The Science of Stories: An introduction to Narrative Psychology. Routledge, London – New York (2008)
5. Lendvai P., Ehmann B., Fritz A., László J.: Automatikus eljárás szemantikus szerepek narratív pszichológiai vizsgálatára. A tudomány emberi arca – A Magyar Pszichológiai

Társaság XXI. Országos Tudományos Nagygyűlése Nyugat-magyarországi Egyetem, Szombathely, 2012. május 30-június 1. (2011)
6. Miháltz, M.: Knowledge-based Coreference Resolution for Hungarian. In: Proceedings of The Sixth International Conference on Language Resources and Evaluation. Marrakesh, Morocco (2008)
7. Prószéky G., Miháltz M.: Magyar WordNet: az első magyar lexikális szemantikai adatbázis. Magyar Terminológia, Vol.1 (2008) 43–57
8. Vincze O., Gábor K., Ehmann B., László J.: Technológiai fejlesztések a NooJ pszichológiai alkalmazásában. In: Tanács A., Szauter D., Vincze V. (szerk.): VI. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2009) 285–294

A Regresszív Képzeleti Szótár magyar nyelvű változatának létrehozása

Pólya Tibor[1], Szász Levente[1, 2]

[1] MTA TTK Kognitív Idegtudományi és Pszichológiai Intézet
1132 Budapest, Victor Hugó utca 18-22.
polya.tibor@mta.ttk.hu
[2] Pécsi Tudományegyetem, Pszichológiai Intézet
7624 Pécs, Ifjúság útja 6.
levente.szasz@mtapi.hu

Kivonat: A Regresszív Képzeleti Szótár az egyik legelterjedtebben használt automatikus pszichológiai szövegelemző eljárás. Az előadás a szótár magyar nyelvű változatának elkészítési folyamatát mutatja be. A magyar nyelvű szótár megbízhatóságának mérését Wilson [17] eljárása alapján végeztük el. Eredményeink azt mutatják, hogy a Regresszív Képzeleti Szótár magyar nyelvű változata megbízható mérési eszköz.

1 A Regresszív Képzeleti Szótár

A Regresszív Képzeleti Szótárt – amelynek eredeti neve Regressive Imagery Dictionary, röviden RID – Colin Martindale [5] hozta létre angol nyelven. A RID a legismertebb pszichológiai tartalomelemző eljárások közé tartozik. Népszerűségét mutatja az is, hogy az elmúlt négy évtized során öt nyelvre fordították le [9]. A RID az elsődleges és a másodlagos gondolkodási folyamatokra utaló tartalmakat azonosítja a szövegben. Az elsődleges gondolkodási folyamatra az jellemző, hogy asszociatív, konkrét és a realitáshoz kevésbé kapcsolódó [12]. A fantázia, az ábrándozás és az álmok fő gondolkodási módja [10]. Ezzel szemben a másodlagos gondolkodási folyamat absztrakt, logikus, realitás központú és problémamegoldásra fókuszáló [12].

A RID az elsődleges és másodlagos gondolkodási folyamatokhoz kapcsolódó tartalmakat a szavak szintjén azonosítja, amihez hierarchikusan szervezett szótárakat használ. A hierarchikus szerveződésnek két csúcskategóriája van: az elsődleges gondolkodási folyamatra utaló szavak szótára, amely 1828 szócsonkot tartalmaz és a másodlagos gondolkodási folyamatra utaló szavak szótára, amely 714 szócsonkot foglal magában. (A két csúcskategóriát Martindale a későbbiekben kiegészítette egy érzelmi szótárral is. Ez azonban elméletileg nem kapcsolódik a gondolkodási mód fogalmához, így ezt a szótárt nem fordítottuk le. Ugyanakkor rendelkezésre áll magyar nyelvű érzelmi szótár [4].) Az *Elsődleges gondolkodási folyamatok szótára* 5 kategóriára bomlik, ezek a kategóriák a következő szinten 29 alszótárt foglalnak magukban

(lásd 1. táblázat). A *Másodlagos gondolkodási folyamatok szótára* 7 alszótárból áll, és nem tartalmaz köztes szintet (lásd 2. táblázat).

1. táblázat: Az elsődleges gondolkodási folyamat kategóriái angol és magyar nyelvű példákkal.

Kategória		**Angol nyelvű példák**	**Magyar nyelvű példák**
Drive	**Oralitás**	Breast, drink, lip	Mell, iszik, ajak
	Analitás	Sweat, rot, dirty	Izzad, rohad, piszkos
	Szex	Lover, kiss, naked	Szerető, csók, meztelen
Érzékelés	**Általános érzékelés**	Fair, charm, beauty	Tetszetős, báj, szépség
	Érintés	Touch, thick, stroke	Érintés, sűrű, cirógat
	Íz	Sweet, taste, bitter	Édes, íz, keserű
	Szag	Breath, perfume, scent	Lehelet, parfüm, illat
	Hang	Hear, voice, sound	Hall, hang, zörej
	Látvány	See, light, look	Lát, fény, néz
	Hideg	Cold, winter, snow	Hideg, tél, hó
	Kemény	Rock, stone, hard	Szikla, kő, kemény
	Lágy	Soft, gentle, tender	Lágy, enyhe, puha
Védekezés	**Passzivitás**	Die, lie, bed	Meghal, fekszik, ágy
	Utazás	Wander, desert, pilgrim	Vándorlás, sivatag, zarándok
	Random mozgás	Wave, roll, spread	Hullám, gurul, terjed
	Diffúzió	Shadow, cloud, fog	Árnyék, felhő, köd
	Káosz	Wild, crowd, jungle	Vad, tömeg, dzsungel
Regresszió	**Ismeretlen**	Secret, mystic, unknown	Titok, misztikus, ismeretlen
	Időtlen	Eternal, forever, immortal	Örök, örökké, halhatatlan
	Tudatváltozás	Dream, sleep, wake	Álom, alszik, ébred
	Áthaladás	Road, wall, door	Út, fal, ajtó
	Nárcizmus	Eye, heart, hand	Szem, szív, kéz
	Konkrétság	Here, behind, west	Itt, mögött, nyugat
Ikaroszi képzelet	**Emelkedés**	Rise, fly, throw	Emelkedik, repül, eldob
	Magasság	Airplane, bird, tower	Repülőgép, madár, torony
	Esés	Fall, slide, sink	Zuhan, csúszda, süllyed
	Mélység	Cave, valley, submarine	Barlang, völgy, tengeralattjáró
	Tűz	Fire, flame, smoke	Tűz, láng, füst
	Víz	Sea, water, swim	Tenger, víz, úszik

2. táblázat: A másodlagos gondolkodási folyamat kategóriái angol és magyar nyelvű példákkal.

Kategória	**Angol nyelvű példák**	**Magyar nyelvű példák**
Absztrakció	Know, reason, think	Tud, ok, gondol
Társas	Tell, help, advice	Mond, segít, tanács
Instrumentális	Win, find, work	Nyer, talál, munka
Korlátozás	Arrest, forbid, stop	Letartóztat, tilt, megállít
Rend	List, simple, symmetric	Lista, egyszerű, szimmetrikus
Idő	Yesterday, year, now	Tegnap, év, most
Erkölcs	Law, virtue, responsibility	Törvény, erény, felelősség

A RID pszichológiai validitását számos empirikus vizsgálat eredménye igazolta, amelyeket – többek között – gyerekektől [16], pszichotikus betegektől [14], illetve akut droghatás alatt álló személyektől [15] nyert szövegeken végeztek el. A RID-et gyakran alkalmazzák az irodalmi szövegek alkotásához köthető pszichológiai folyamatok kutatására is [6].

2 A magyar Regresszív Képzeleti Szótár fordításának folyamata

2.1 Döntés a karakteralapú keresés alkalmazása mellett

A RID magyar nyelvű változatát – az angol eredetivel megegyező módon – a karakteres keresés elvén hoztuk létre. Választásunkat két szempont indokolta. Egyrészt a pszichológiai szövegelemzésben a karakteres keresést alkalmazó tartalomelemző szoftverek terjedtek el (például WordStat [2], LIWC [8]). Így a karakteres keresés elvét alkalmazva könnyebben tudjuk kombinálni ezt az elemzési eljárást más elemzési eszközökkel. Másrészt a munka elkezdésekor – 2010-ben – nem állt rendelkezésünkre megfelelő lefedettséget biztosító magyar nyelvű szótár.

2.2 A folyamat fontosabb lépéseinek áttekintése

Az első lépés az úgynevezett nyers fordítás elkészítésének fázisa volt. Ennek során az angol nyelvű RID-ben szereplő szócsonkok alapján összegyűjtöttük azokat a magyar nyelvű szavakat, amelyek angol megfelelőit a RID angol változata találatként azonosítja. Ebben a munkában 10 pszichológus hallgató vett részt.

A második lépésben ezen szavak listájáról a cikk két szerzője kiválogatta azokat a szavakat, amelyek jelentése kapcsolódik a RID valamelyik alszótárához.

Harmadik lépésként előállítottuk azokat a magyar nyelvű szócsonkokat, amelyek a toldalékolástól függetlenül azonosítják az előző lépésben felsorolt szavakat. Az így kapott szócsonkokat találati listákon helyeztük el, amelyeken helyet kaptak többszavas kifejezések is.

2.3 A fejlesztéshez használt program

A szótárépítést a Max Silberztein által megalkotott NooJ [11] számítógépes nyelvi fejlesztő környezete segítségével valósítottuk meg. A NooJ grafikus felületét felhasználva hoztuk létre a gráfokat vagy más néven lokális nyelvtanokat. A szavakat virtuális keretekbe, úgynevezett boxokba helyeztük el. Ezek tetszőleges módon összeköthetőek, így akár több szóból álló, szintaktikai információt is tartalmazó keresőkifejezések is létrehozhatóak.

2.4 A keresés módja

A karakteres keresésnek két módja van. Az alapértelmezett mód a kezdő karaktersor megadása. Ebben az esetben az algoritmus az összes olyan szót megtalálja, amely ezt a feltételt teljesíti. Például a "szép*" karaktersor (*Általános érzékelés kategória*) megadásával a rendszer kinyeri a szövegből a "szépet", "szépnek", "szépről" stb. alakokat is. (A „*" karakter azt jelöli, hogy a szócsonk tetszőleges karakterrel/karakterekkel folytatódhat.)

Figyelembe vettük azt is, hogy bizonyos lexémák változó tővel rendelkeznek. Emiatt például az „alma" (*Oralitás kategória*) szó esetében az „almá*" karaktersort is felvettünk a listára, hogy többek között a birtokos személyjellel ellátott „almám", valamint a tárgyas „almát" alakokat is felismerje a rendszer.

A kettős mássalhangzóra végződő szavaknál úgy kellett megadnunk a kezdő karaktersort, hogy a –val, –vel ragos hasonult alakokat is megtalálja a kereső algoritmus. Például „kalács*" (*Oralitás kategória*) helyett „kalác*" gráfba építésére volt szükség a „kaláccsal" szóalak megtalálása érdekében.

A karakteres keresés második módja a pontos karaktersor megadása, amely csak a teljes mértékben egyező karaktersorból álló szóalakra ad találatot. Ezt alkalmaztuk például az "itt" (*Konkrétság kategória*) határozószó felismeréséhez. A kettőnél több tővel rendelkező igéknél is egyszerűbbnek bizonyult az összes ragozott alak pontos bemásolása az adott gráfba ahhoz képest, mintha például az „eszik" (*Oralitás kategória*) ige „esz-", „ev-"„e-", „é-", „en-" töveit adtuk volna meg kezdő karaktersorként, mivel így nagyon sok téves találat keletkezett volna.

A kezdő karaktersorral való azonosítást jóval gyakrabban alkalmaztuk, mint a pontos karaktersorral való azonosítást.

2.4.1 A találatok és kizárások

Találati listák létrehozása mellett készítettünk olyan listákat is, amelyeket a NooJ kizár az elemzésből. Például a "menta*" (*Oralitás kategória*) kezdő karaktersor megadásával kinyerésre kerül a szövegből a "mentalevél" szó, ami beletartozik az *Oralitás kategóriába,* azonban a "mentalitás" és "mentalista" szavak is találatként jelentkeznek, holott ezek nem tartoznak bele ebbe a kategóriába. Ezért az utóbbiakat felvettük a kizárási listára, amit az '+EXCLUDE' „tag" használatával valósítottunk meg.

Minden egyes szócsonk esetén az összes lehetséges téves találatot számításba vettük. Ezt az ELRAGOZ (Elektronikus magyar ragozási szótár [3]) programnak az a funkciója tette lehetővé, amely valamennyi olyan szót kilistáz (a szoftver memóriájá-

ban tárolt 73810 címszó közül), amely a felhasználó által megadott karaktersorral kezdődik. A „nyer*” (*Instrumentális kategória*) karaktersor esetén a listába kerül például a „nyers” és a „nyereg” szó is.

2.5 Az igekötős igék kezelése

Ha egy adott ige és a belőle származtatható összes igekötős alak adekvátnak számított egy adott alszótár szempontjából, akkor felsoroltuk az összes olyan esetet, ahol az igekötő az ige előtt áll – vele egybeírva. Például „besegít”, „átsegít”, „kisegít” (*Társas kategória*). Ezeken túl csak magát az igét kellett megnevezni („segít”), amelynek megadásával egyúttal a fordított sorrendű változatok is (például: „segít be”, „segít át”) megtalálására kerülnek a gráf lefuttatásakor.

Amennyiben azonban az adott ige csak bizonyos igekötőkkel képez találatot, másokkal együtt állva pedig kategórián kívülinek minősül, akkor magának az igének (például „dönt” [*Absztrakció kategória*]), valamint az ’igekötő az ige előtt áll’ formáknak (például: „eldönt”) a gráfban történő feltüntetésén túl az is szükséges volt, hogy az adott alszótár szempontjából nem odaillő, fordított sorrendű változatokat, például a „dönt fel” kifejezést kizárjuk.

2.6 Az azonos alakú szavak esete

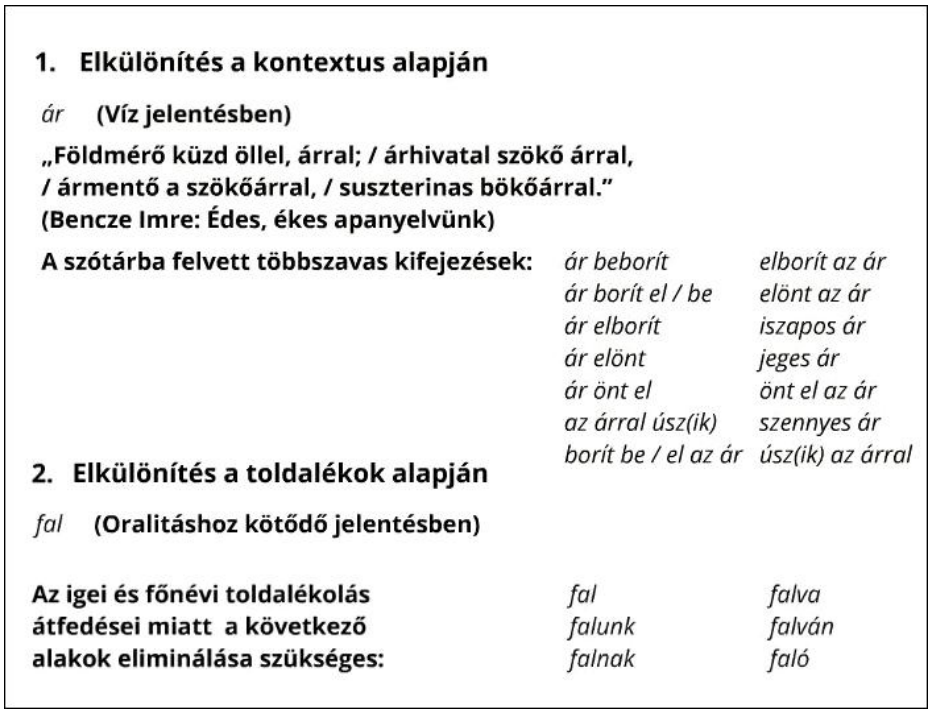
A karakteres kereső algoritmusok létrehozásakor az egyik leginkább időigényes folyamatot az azonos alakú, találati és téves találati minőségben egyaránt előforduló szavak elkülönítése jelentette. Ezekben az esetekben leggyakrabban az ige és a névszó differenciálására volt szükség.

A kiindulást minden esetben az jelentette, hogy a Magyar Nemzeti Szövegtár [13] korpusznyelvészeti adatbázis segítségével felmértük a találati és a téves találati előfordulások gyakoriságát. Ezekre az adatokra támaszkodva hoztuk meg a döntésünket arra vonatkozólag, hogy szerepeltessük-e az adott karaktersort a szótárban, és amennyiben igen, akkor milyen módon végezzük az egyértelműsítést. Erre mutatunk az alábbiakban két példát.

Az elkülönítés egyik módja a kontextus figyelembevételével történt. Ebben az esetben több szóból álló kifejezéseket használtunk fel az azonosításhoz. Például az „ár” szónak a *Víz kategória* szempontjából adekvát jelentésén kívül más használatai is ismeretesek (lásd az 1. ábrán szereplő idézetet). Emiatt magának az „ár” karaktersornak a találati listára való felvétele helyett kizárólag az 1. ábrán szereplő kifejezéseket szerepeltettük a gráfban.

Másik lehetőség a toldalékok alapján történő elkülönítés volt. A „fal” (eszik) igeként az *Oralitás kategóriába* tartozik, főnévként (épület része) azonban nem képezi részét sem ennek az alszótárnak, sem más alszótárnak. Annak érdekében, hogy az alaki egybeesés ellenére – adekvát jelentésben – szerepelhessen a kategóriában, az ELRAGOZ program segítségével kilistáztuk a „fal” szó toldalékolt alakjait mind az igei, mind a főnévi előfordulás szerint. Eliminátuk azokat a szóalakokat (lásd 1. ábra), amelyek egybeesést mutattak. Ez 3 eset törlését jelentette, a többi igealakot, ami-

ből 56 volt, feltüntethettük a szótárban. Továbbá eltávolításra került három igenév is, amelyek két másik főnév (falu és faló) meghatározott alakjaival voltak azonosak.

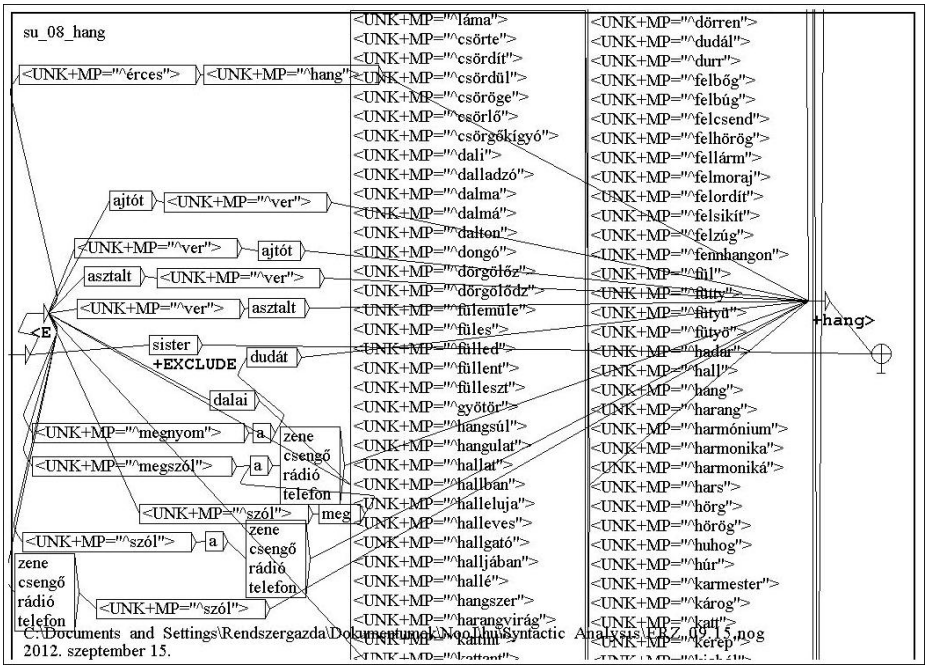


1. ábra: Példák az azonos alakú szavak elkülönítésének lehetőségeire.

A magyar nyelvű változat *Elsődleges gondolkodási folyamat szótára* 4521 karaktersort és 260 két vagy több karaktersorból álló kifejezést tartalmaz. A *Másodlagos gondolkodási folyamat szótár* 2020 karaktersorból és 1098 kifejezésből áll. A kizárási listán 1785 karaktersor, illetve kifejezés szerepel. Az egyik alszótár, a *Hang* gráfjának részlete a 2. ábrán látható.

2.7 A magyar Regresszív Képzeleti Szótár a WordStat rendszerében

A WordStat [2] kereskedelmi forgalomban megvásárolható, tartalomelemzésre és szövegbányászatra alkalmas szoftver. A Wordstat a RID összes nyelvi változatának használatát lehetővé teszi. A NooJ programmal létrehozott szótárunkat áthelyeztük erre a platformra. Ez a folyamat viszonylag kevés erőfeszítést igényelt; a 2 évig tartó fejlesztés idejének töredékét tette csak ki. A magyar RID a WordStat honlapján is elérhető, illetve használható. (Azok, akik szeretnék a magyar nyelvű RID-et elemzésre használni, szövegeiket közvetlenül a szerzőknek is elküldhetik.)



2. ábra: A Hang alszótár gráfjának részlete.

3 A reliabilitás vizsgálata

A magyar nyelvű RID reliabilitásának megállapítása az elsődleges és másodlagos gondolkodási folyamatok szintjén történt. A beméréshez Wilson [17] eljárását követtük, aki a RID portugál, latin és német nyelvű fordításainak megbízhatóságát vetette össze, gold standardként az eredeti, angol verziót használva. Elgondolása szerint a reliabilitás mértékét az mutatja meg, hogy mennyire őrzi meg az adott fordítás az elsődleges és másodlagos tartalmak egymáshoz viszonyított arányát. A Biblia 150 zsoltárán végezte el az elemzést. Elemzésében a zsoltárokat külön egységként kezelte. Az angol nyelvű RID-et a Challoner által revideált Douay-Rheims-féle bibliafordításon [1] futtatta le.

Wilson minden egyes zsoltárt az alábbi 3 csoport valamelyikébe sorolta be: 1. Az elsődleges folyamat domináns 2. A másodlagos folyamat domináns 3. Egymáshoz képest egyik szókategória sem domináns. A dominancia azt jelenti, hogy 5%-os szinten szignifikáns eltérés mutatkozik az eloszlások egyenlőségéhez képest.

Ezt követően páronként végzett összehasonlítást: mindig az angol nyelvű zsoltárhoz hasonlítva a másik nyelvű verziót. Ebben az összehasonlításban ötféle konstelláció lehetséges: 1. Helyes azonosítás: az adott zsoltár angol, illetve más nyelvű változatában azonos módon vagy az *Elsődleges* vagy a *Másodlagos gondolkodási folyamat*

kategória domináns. 2. Helyes elutasítás: az angol és a másik nyelvű szövegre is igaz, hogy egyik kategória sem domináns. 3. Helytelen azonosítás: az angol zsoltárban egyik kategória sem domináns, azonban a másik nyelven a szöveg szignifikáns eltérést mutat akár az elsődleges, akár a másodlagos kategória előfordulásának irányában. 4. Helytelen elutasítás: az angol zsoltárban domináns az elsődleges vagy a másodlagos tartalom, azonban a másik nyelvű verzióban nincs domináns kategória. 5. Fordított azonosítás: mind az angol, mind a másik változatnál jelentkezik dominancia, azonban ezek éppen ellentétes irányúak: ha az angolnál az elsődleges kategóriából van több, akkor a másiknál a másodlagosból, vagy fordítva. A fenti öt pártípus abszolút gyakoriságait fordításonként összesítve Wilson a 3. táblázatban található eredményeket kapta.

A magyar nyelvű zsoltárok elemzését a Káldi György által fordított Szentírás [7] szövegének felhasználásával készítettük el. A magyar nyelvű RID-re vonatkozó adatokat a 3. táblázat utolsó oszlopa tartalmazza. A reliabilitás méréséhez Wilson nyomán a következő mutatókat használtuk fel. 1. Pontosság (accuracy): A helyesen (vagyis az angol változattal megegyezően) kategorizált szövegek arányát adja meg az összes szöveg számához viszonyítva. 2. Érzékenység (sensitivity): A helyes azonosítások arányát mutatja azokban az esetekben, amikor az angol szövegben valamelyik kategória domináns. 3. Specifikusság (specificity): A helyes elutasítások arányát mutatja azokban az esetekben, amikor az angol szövegben egyik kategória sem domináns.

3. táblázat: A zsoltárszövegek konstellációinak gyakoriságai
(Wilson [17] adatainak felhasználásával)

A konstelláció típusa	**Portugál**	**Latin**	**Német**	**Magyar**
Helyes azonosítás	27	25	14	24
Helyes elutasítás	58	91	88	92
Helytelen azonosítás	53	20	23	19
Helytelen elutasítás	12	14	25	15
Fordított azonosítás	0	0	0	0

A magyar nyelvű Regresszív Képzeleti Szótár megbízhatóságára vonatkozó eredményeket a 4. táblázat utolsó oszlopa mutatja. Látható, hogy a magyar fordítás két mutató tekintetében ért el az összehasonlított nyelvi változatok között első helyezést (egyik ezek közül holtverseny), egy esetben pedig harmadik helyezést a négy közül. Ez alapján megállapítható, hogy a magyar fordítás megbízhatóan használható az elsődleges és másodlagos gondolkodási folyamatokhoz kapcsolódó tartalmak mérésére.

4. táblázat: A RID-fordítások reliabilitás mutatói
(Wilson [17] adatainak felhasználásával)

A megbízhatóság mutatói	**Portugál**	**Latin**	**Német**	**Magyar**
Pontosság	56,67 %	77,33 %	68 %	77,33 %
Érzékenység	69,23 %	64,1 %	35,9 %	61,54 %
Specifikusság	52,25 %	81,98 %	79,28 %	82,88 %

Hivatkozások

1. Challoner's revised Douay-Rheims Version Old Testament (1609–1610) The Whole Revised and Diligently Compared with the Latin Vulgate by Bishop Richard Challoner (1749-1752). Letöltve: http://www.gutenberg.org/cache/epub/1610/pg1610.html, Letöltés időpontja: 2012. 08. 01.
2. Davi, A., Haughton, D., Nasr, N., Shah, G., Skaletsky, M., Spack, R.: A review of two text-mining packages: SAS TextMining and WordStat. American Statistician, Vol. 59, No. 1 (2005) 89–103. A program elérhetősége: http://provalisresearch.com/products/content-analysis-software
3. ELRAGOZ (Elektronikus magyar ragozási szótár) szoftver. MorphoLogic Kft.
4. Fülöp, É., László, J.: Az elbeszélések érzelmi aspektusának vizsgálata tartalomelemző program segítségével. In: IV. Magyar Számítógépes Nyelvészeti Konferencia. SZTE, Szeged (2006) 296–304
5. Martindale, C.: Romantic Progression: The Psychology of Literary History. Hemisphere, Washington (1975)
6. Martindale, C.: The Clockwork Muse: The Predictability of Artistic Change. Basic Books, New York (1990)
7. Ószövetségi Szentírás a Neovulgáta alapján. Fordította: Káldi György. Szent Jeromos Bibliatársulat, Budapest (1997). Letöltve: http://www.biblia-tarsulat.hu/bibliaszoveg.htm. Letöltés időpontja: 2012. 08. 03.
8. Pennebaker, J. W., Francis M. E., Booth, R. J.: Linguistic Inquiry and Word Count (LIWC): LIWC2001. Lawrence Erlbaum Associates, Mahwah (2001)
9. RID különböző nyelvű moduljainak frissített listája az alábbi webcímen érhető el: . http://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/ regressive-imagery-dictionary-by-colin-martindale-free/
10. Russ, S. W.: Primary process thinking and creativity: Affect and cognition. Creativity Research Journal, Vol. 13 (2001) 27–35
11. Silberztein, M.: Nooj Manual. (2003) Letöltve: http://www.nooj4nlp.net/NooJManual.pdf Letöltés időpontja: 2012. 08. 02.
12. Suler, J. R.: Primary process thinking and creativity. Psychological Bulletin, Vol. 88. (1980) 144–165
13. Váradi T.: The Hungarian National Corpus. In: Proceedings of the 3rd LREC Conference, Las Palmas, Spanyolország (2002) 385–389. Elérhetőség: http://corpus.nytud.hu/mnsz
14. West, A. N., Martindale, C.: Primary process content in paranoid schizophrenic speech. Journal of Genetic Psychology, Vol. 149 (1988) 547–553
15. West, A. N., Martindale, C., Hines, D., Roth, W.: Marijuana-induced primary process content in the TAT. Journal of Personality Assessment, Vol. 47 (1983) 466–467
16. West, A. N., Martindale, C., Sutton-Smith, B.: Age trends in the content of children's spontaneous fantasy narratives. Genetic, Social, and General Psychology Monographs, Vol. 111. (1985) 389–405
17. Wilson, A.: The Regressive Imagery Dictionary: A test of its concurrent validity in English, German, Latin, and Portuguese. Literary and Linguistic Computing, Vol. 26, No. 1 (2011) 125–135

V. Morfológia, szintaxis

Helyesírás.hu – Nyelvtechnológiai megoldások automatikus helyesírási tanácsadó rendszerben

Miháltz Márton[1], Hussami Péter[2], Ludányi Zsófia[1], Mittelholcz Iván[1], Nagy Ágoston[3], Oravecz Csaba[1], Pintér Tibor[1], Takács Dávid[1]

[1] MTA Nyelvtudományi Intézet, Nyelvtechnológiai Osztály, 1068 Budapest, Benczúr u. 33.
{mihaltz.marton, ludanyi.zsofia, mittelholcz.ivan, oravecz.csaba, pinter.tibor}
@nytud.mta.hu,takdavid@gmail.com

[2] Rényi Alfréd Matematikai Kutatóintézet, 1053 Budapest, Reáltanoda utca 13–15.
hussami@renyi.hu

[3] Szegedi Tudományegyetem, Bölcsészettudományi Kar, Francia Nyelvi és Irodalmi Tanszék, 6722 Szeged, Egyetem u. 2.
nagyagoston@lit.u-szeged.hu

Kivonat: A helyesiras.hu projekt célkitűzése egy nyelvtechnológiai eszközökkel támogatott magyar helyesírási tanácsadó portál kifejlesztése, mely 7 különböző területen próbál interaktív segítséget nyújtani: külön- és egybeírás, helyesírás-ajánló, elválasztás, tulajdonnevek írása, számnevek helyesírása, keltezés, betűrendbe sorolás. A cikkben szeretnénk bemutatni a webalkalmazások kifejlesztésében alkalmazott nyelvtechnológiai erőforrásokat és algoritmusokat, valamint a leginkább problémás kérdést jelentő témakör, a különírás-egybeírás fejlesztése során felmerült kihívásokat. A magyar helyesírás rendszere nagymértékben épít a nyelvhasználók értelmezési képességeire, így ahol lehet, a rendszer többféle választási lehetőséggel, visszakérdezésekkel próbálja a hiányzó egyértelműsítő információkat beszerezni.

1 Bevezetés

A magyar helyesírás számítógépes, illetve formális igényű feldolgozása nem új keletű a magyar nyelvészetben [1], [3], [10]. A helyesírási szabályok komplexitása, „túlszabályozottsága" és inkonzisztenciája miatt azonban számítógéppel feldolgozható teljes formális leírást nem sikerült kidolgozni. Az MTA Nyelvtudományi Intézete 2009 óta dolgozik egy olyan szakértői rendszeren, amelyben nyelvtechnológiai eljárások alkalmazásával lehetővé válik a helyesírási kérdésekre automatikus választ adni – esetenként a kérdezők aktív bevonásával (a kezdeti lépésekről, tervekről, illetve a megvalósítás nehézségeiről bővebben lásd [6].) A *helyesírás.hu* portál célja a tudatos és kevésbé tudatos helyesírók hasznos segédeszközévé válni: egyrészt a válaszok gyorsaságával, másrészt megbízhatóságával. A magyar helyesírás túlszabályozottságát és kiterjedt ismeretigényét tekintve nyilvánvaló, hogy a rendszer nem tud minden szabálypontot tökéletesen kezelni. Az azonban elvárható, hogy tisztában legyen a „gyenge pontja-

ival", azaz csak azokra a kérdésekre válaszoljon, amelyekre valós találatot és szabálypontot talál – ellenkező esetben a rendszer visszakérdez, bevonva a kérdezőt a helyes alak kiválasztásába (intuitív döntések meghozatala), illetve végső esetben felkínálja a humán szakértői segítség lehetőségét.

A portál gerincét alkotó dinamikus alkalmazások mellett hangsúlyt kapnak a statikus tartalmak is, melyek segítik a helyesírásban tájékozódni vágyókat. A portálra látogatók megtalálják az Akadémiai Kiadó Magyar helyesírás szabályainak (AkH.) [7] mindenkori legújabb kiadását, valamint többféle megközelítésben böngészhetik az MTA Nyelvtudományi Intézetének nyelvi tanácsadó szolgálatán gyűjtött kérdéseket és az azokra adott válaszokat (kategóriacímkék és szakmai besorolás címkék alapján történő böngészés, valamint különböző kategóriák szerint kezelhető szabadszavas keresés formájában).

A dolgozat részletesen beszámol az AkH. (azaz az akadémiai helyesírás) alkalmas szabálypontjainak formalizálásáról, a rendszer egyes moduljainak működéséről, a modulokat működtető szabályrendszerek felépítéséről, valamint kitér a rendszer mögött álló speciális lexikonok kapcsolódásának és integrálásának módjára.

2 A rendszer általános felépítése

A felhasználói kérdések hatékony és automatikus kezelhetőségének érdekében két alapvető és egymással ellentétes cél között kell ergonomikus kompromisszumot találni: (1) kényelmes, a felhasználótól a lehető legegyszerűbb bemenetet igénylő felület; (2) a várható inputról minden lehetséges forrásból a lehető legtöbb információt begyűjtő rendszer. A minden megszorítás nélküli, gyakorlatilag tetszőleges szöveg bevitelét megengedő bemenet automatikus feldolgozása lehetetlen vállalkozás, ezért első lépésben feltétlen szükséges minimálisan a kérdéskörök, problémakategóriák egyértelmű meghatározása, és ennek alapján a választ előállító feldolgozási lépések kiválasztása.

Az aktuális helyesírási problémakört (kategóriát) a felhasználó választja ki a felületen megjelenő kötelező típusválasztó segítségével. A kiválasztás függvényében rendelődik a bemenethez egy meghatározott feldolgozó modul, mely a választ előállítja. Ezek a modulok alapvetően két csoportba sorolhatók:

1. A kategóriaválasztás alapján olyan meghatározott problémakört kezelő modulok, melyek csak specifikus, megszorított bemenetet fogadnak és erre egyértelmű választ adnak. Ilyenek a számnevek helyesírása, keltezés, betűrendbe sorolás (3., 4. és 5. rész), tulajdonnevek írása (6. rész) és az elválasztás (7. rész) kategóriákat kezelő modulok.

2. Gyakorlatilag szabadszöveges (bár a lehetőségekhez mérten szűrt) bemenetet kezelő feldolgozó modulok: a helyesírás-ajánló (8. rész) és a külön- és egybeírást kezelő modul (9. rész). Ez utóbbi a legkomplexebb, egyben a legaktívabb felhasználói interakciót is igénylő komponens, ahol az elérhető információ maximalizálása alapvető fontosságú.

A helyesírási kérdések jelentős részének megválaszolása egy bemeneti karaktersorozatban rendelkezésre álló információ mellett további, a közvetlen felszíni jellemző-

kön messze túlmutató ismereteket is igényel. Ezeket vagy erőforrásként a rendszerhez kell csatolni, vagy adott esetben a felhasználótól megszerezni. A rendszer erőforrásai egyrészt a **Humor** morfológiai elemző ([5], [8]), másrészt olyan lexikális adatbázisok, melyek a helyesírási szempontból speciálisan viselkedő lexikális elemek lehetőleg kimerítő felsorolását tartalmazzák (színnevek, anyagnevek, mindig egybeírandó előtagok, egybeírt alanyos/tárgyas összetételek stb.). Ezeket az erőforrásokat jórészt a 2. típusú feldolgozó modulok használják, illetve a morfológiai elemzőkre támaszkodik még az elválasztást kezelő komponens is.

Az alábbi fejezetekben részletesen bemutatjuk az egyes témaköröket megvalósító webalkalmazásokat működtető modulok célját és felépítését.

3 Számnevek helyesírása

Felhasználói bemeneteként előjel, számjegyek, tizedesvessző és törtvonal valamilyen értelmes kombinációja szolgál. A modul kezeli a tőszámneveket, a sorszámneveket, tizedes- és egyszerű törteket és a felhasználónak visszaadja a megadott szám betűkkel való lehetséges átírásait, kiegészítve ezt az AkH. vonatkozó pontjaira mutató hivatkozásokkal és az esetleges további megjegyzésekkel.

Ahol több helyes átirat lehetséges, ott a rendszer magyarázatokkal igyekszik a felhasználók segítségére lenni. Az egyik ilyen eset a törtek egybe-, illetve különírása. A *2/3* például jelzői értelemben egybe írandó (*kétharmad csésze liszt*), minden egyéb használatban viszont külön (*két harmad nagyobb, mint egy harmad*).

Ehhez hasonló eset a *kettő* és a *két* megfelelő használata. Jelzői értelemben a *két* használata az elfogadott (*két pár zokni*), míg a *kettő* csak nem-jelzői értelemben felel meg a standard nyelv- és íráshasználatnak (*egy meg egy az kettő*). Ennek megfelelően a *kettő*-nek egy számnév elején vagy a belsejében történő használata egyértelműen szubsztenderd. Így a *harminckettő* elfogadott alaknak számít, szemben a *kettőezer*-rel, vagy a *háromezer-kettőszáz*-zal. A modul megadja az összes lehetséges, helyesen írt alakot (*kétezer* és *kettőezer*), ugyanakkor megjegyzésben hívja fel a felhasználó figyelmét a standard nyelvváltozattól való eltérésre.

4 Keltezés

A keltezés modul (*Dátumok*) *éééé-hh-nn* formátumú bemenetet dolgoz fel. A beolvasás során a formátumnak való megfelelésen túl ellenőrzi a dátum hozzávetőleges helyességét is. Így például kiszűri a március 50-ét éppúgy, mint az április 31-ét. Az ellenőrzés azonban nem teljes körű, nem terjed ki a szökőévek kezelésére, de az olyan történeti eseményeket sem veszi számításba, mint amilyenek a Gergely-naptár bevezetésekor kimaradó napok. Az ellenőrzés ilyen korlátozottsága azonban nem ok nélküli, hiszen ezeket a napokat akkor is le kell tudnunk helyesen írni, ha éppen azt akarjuk kifejezni, hogy nem is léteztek (pl. „*1582. október 10*. egy nem létező dátumot/napot jelöl.”).

A modul két módon segíti a felhasználót. (1) Felsorolja a dátumnak az AkH. által elfogadott írásmódokat (pl. *1582. október 10., 1582. okt. 10., 1582. X. 10.* stb.). (2) Példákat ad a dátum különböző (pl. toldalékolt) használataira: *1582. október 10-én, 1582. október 10-e óta, 1582 októberében* stb.

A megvalósítás során hónapok és számok (napok) listáját használja. Hangrendjük szerint csoportokra bontva ezekből generálja az általa nyújtotta ajánlásokat. Mint a többi modul esetében, a kimenet mellett az AkH. vonatkozó bekezdéseit is visszaadja.

5 Betűrendbe sorolás

A modul célja a felhasználó által megadott latin betűs (de nem feltétlenül csak a magyarban használt betűket tartalmazó) tételeknek a szabályzat szerinti betűrendbe sorolása (az AkH. 14–15. pontjai szerint. A 16. pont által említett kivételes betűrendbe sorolási eseteket, továbbá a számokat is tartalmazó tételeket, és más ábécék szerinti rendezést itt nem kezeljük).

Az általánosan használt szoftverekben többnyire a klasszikus „lexikális” rendezési algoritmus használatos: a két sztring összehasonlítása karakterenként (írásjelekkel stb. együtt), balról kezdve, és az első különböző karakterpár összehasonlítása adja a két sztring összehasonlításának eredményét. Ha az egyik sztring teljes egészében a másik elejét alkotja, akkor ez utóbbit tekintjük a másodiknak.

A szabályzat szerinti rendezés implementálásához több előfeldolgozási lépés is szükséges.

1) A szövegből ki kell válogatni az összehasonlítható (azaz szóalkotó) karaktereket, és a továbbiakban csak ezzel kell dolgozni (AkH. 14. e.).

2) A magyarban nem használt betűket normalizálni kell, hogy azokkal összehasonlíthatók legyenek. Ezeket a magyarban használt egyjegyű betűkre kell leképezni, mert a megfeleltetettjeikkel azonos súllyal veendők figyelembe a rendezéskor (AkH. 15.).

3) A hosszú magánhangzókat a rövid megfelelőjükkel kell azonos értékűnek tekinteni, ugyanakkor máshogy kell kezelni a szóalakokat, ha alakjuk ékezettelenítve megegyezik (AkH. 14. d.). Ugyanez vonatkozik az idegen mellékjeles betűkre is (AkH. 15.).

4) Az összetett (több karakterből álló, mássalhangzókat jelölő) betűket azonosítani kell, mert ezek egy egységként kezelendők, a hosszúakat pedig két rövid megfelelőjükkel kell helyettesíteni (AkH. 14. c.). Ennek a feladatnak a korrekt megoldása szótárat, illetve morfológiai elemzőt igényel, hiszen sok esetben a többjegyű betűnek is tekinthető karaktersorban morfémahatár van.

Az így előállított betűsorokra már alkalmazható a klasszikus rendezési algoritmus.

6 Tulajdonnevek helyesírása

A tulajdonnevek helyesírásának ellenőrzését a Névkereső modul támogatja. A tulajdonnevek közötti böngészést azok nagy száma miatt (jelenleg több mint kétszázezer) nem tesszük lehetővé, de a nevek kereshetők, és a szűkített találati listákat már megje-

lenítjük. Amint a felhasználó elkezdi begépelni a tulajdonnév első néhány karakterét, a modul prediktíven megjeleníti az illeszkedő tulajdonnevek listáját. Az egyes tulajdonnevek mellett azok besorolása is megjelenik, úgymint pl. földrajzi név, személynév stb.

A személynevek és a földrajzi nevek esetében a névalakok mellett további jegyeket is megjelenítünk (vezetéknév/keresztnév, férfi/női vagy becenév, illetve településnév; közterület neve, magyarországi vagy nem).

Az adatbázis legfontosabb forrásai: a *hunmorph*, a *huntag* és a *Hunspell*[1] szabadon letölthető erőforrásai, a Magyar Posta nyilvános listái, a publikusan hozzáférhető telefonkönyvi adatok, továbbá a FÖMI és az egyes minisztériumok által közölt névtárak és listák. A földrajzi nevek tára elsősorban a magyar vonatkozású neveket tartalmazza. Ez az adathalmaz így is több mint százezer nevet tartalmaz. A vezetéknevek száma szintén több mint százezer elemű, amhez több mint hatezer kereszt- és becenév járul. Ez az adathalmaz is elsősorban a magyar névtárakban és korpuszokban előforduló neveket foglalja magába.

Terveink között szerepel az oktatási, kulturális, civil és egyházi szervezetek neveinek, továbbá városrészek, településrészek neveinek felvétele az adatbázisba úgy, hogy egyes elemeik szerint is és egészükben is kereshetők és listázhatóak legyenek.

Speciális esetet képeznek a cégnevek. A hivatalos cégjegyzékben szereplő cégnevek közt nagy számban szerepelnek hibás alakok is – a nyilvánvaló elgépeléstől kezdve az egyes szavak jellegzetes helyesírási hibáin át az értelemzavaró, tévesztéses hibákig. Hivatalos kontextusban a cégneveket a bejegyzett alakjukban kell használni, akkor is, ha egyébként hibásnak minősülnek, tehát a javított alakok használata nem javasolható egyértelműen. Ezt a diszkrepanciát elkerülendő, jelenleg a cégneveket nem tartalmazza a rendszer mögött álló adatbázis.

7 Elválasztás

Az elválasztást leginkább tipográfiai okok miatt alkalmazzák. Célja, hogy egy szöveg minden sorában a sorok (kézírás esetén) és a szóközök egyenletes nagyságúak, utóbbiak minél rövidebbek legyenek.

Mivel számos szövegszerkesztő alkalmazás használ elválasztómodult, kézenfekvő volt a *helyesírás.hu* projektben is ilyenre támaszkodni, mégpedig az *OpenOffice/LibreOffice* irodai programhoz létrehozott **huhyphn-**re.

A huhyphn egy nyelvspecifikus karaktersorozatokat tartalmazó fájlt használ, melyben a tiltott és a lehetséges elválasztási helyek számokkal vannak jelölve: az előbbiek párossal, az utóbbiak páratlannal. A hosszabb karaktersorozat és a nagyobb szám felülírja a kisebbet.

A magyarban például az *en1d2ő* szabály lehetővé teszi az elválasztást az *n* és a *d* között és tiltja a *d* és az *ő* között, így lesz *ken-dő* vagy *kerülen-dő*, de sosem *kend-ő*. Ezt a szabályt felülírja a *ren2d3őr*, ami által a *rendőr* szó elválasztása *rend-őr* lesz (lévén összetett szó), miközben a *ken-dő* továbbra sem változik [4].

1 http://mokk.bme.hu/en/eszkozok/

A huhyphn-t sok esetben módosítani kellett, mivel főleg tipográfiai célokra készült, így nem engedélyez például olyan elválasztásokat, mint *a-pa-i*, mivel egy karakter leválasztása nyomdai szövegben nem esztétikus. Ezeket a tiltásokat a már meglévő szabályokban írtuk át: az adott helyen a páros számot kicseréltük egy páratlannal vagy új szabályt vettünk fel.

A huhyphn egyik tipikus tulajdonsága, hogy a többféleképpen elválasztható, többjelentésű szavakat nem, vagy csak az egyik lehetséges módon választja el. Ilyen szó például a *megint*, amelyet határozóként *me-gint*, igekötős igeként *meg-int* alakban lehet elválasztani. Az AkH. 233-238. szerint a szóösszetételi határok mentén kell elválasztani, de mivel a huhyphn szóegyértelműsítést nem végez, illetve rontani sem szeretne, ezért az ilyen típusú szavakat nem választja el. Ugyanígy a tanárok szót is csak *ta-ná-rok* alakban választja el, holott ezt *tan-á-rok* formában is lehetséges.

Ezen esetek kiküszöbölésére alkalmaztuk a *Humor* morfológiai elemzőt [5], [8]. Ezzel az alkalmazással minden input szóról el tudjuk dönteni, hogy az összetett-e: ha igen, akkor a szóösszetételeket külön-külön választjuk el, majd azokat egyesítjük, és azok közé |- jelet teszünk, például *kis*|*-a-u-tó* vagy *tan*|*-á-rok*. Ha egy szónak több morfológiai elemzése van, akkor mindegyiket számba vesszük, így lesz az *altest* szóból *al*|*-test* vagy *alt*|*-est*. Végül a szóösszetétel-elemzés után a kimenetet egyesítjük a huhyphn standard kimenetével: például akkor, ha a morfológiai elemzést követően az elválasztás *tan*|*-á-rok*, azonban a huhyphn szerint *ta-ná-rok*, akkor a modul mindkettőt visszaadja.

Az alkalmazást jelenleg egy egymillió szavas listán (MNSz gyakorisági lista) tesztteljük: ha a program olyan szótagot talál, amelyben a magánhangzók száma nem egy, akkor azt jelzi, és ha szükséges, azt kézzel javítjuk (a *Mar-seille* esetében például nem kell).

8 Helyesírás-ajánló

A különálló szavak helyesírásának ellenőrzése modul (*Helyes-e így?*) a felhasználó által megadott szóalakok létezését vizsgálja, helytelen alakok esetében javaslatot tesz a leginkább hasonló helyes alakokra. Ebben a modulban kombináljuk a nyílt forrású **Hunspell**[2] helyesírás-ellenőrző és a MorphoLogic **Humor** morfológiai elemzőjére [5], [8] építő helyesírás-ellenőrző motorok kimenetét.

A Hunspell 1.3.2-es verziója a *MySpell* motorjára épül, de támogatja a szóösszetételeket és a gazdag morfológiájú nyelveket is, a szabadon hozzáférhető **Magyar Ispell**[3] szótár 1.6.1-es verziójával.

A MorphoLogic helyesírás-ellenőrzője azokat a szóalakokat fogadja el helyesként, amelyeket a benne működő Humor morfológiai elemző képes a tárolt morfémákból a nyelvi szabályoknak eleget téve összerakni (kb. 100 000 alapszó több mint kétmilliárd szóalakja).

Az ismeretlen szavak nagy részéhez mindkét motor képes javítási javaslatok listáját visszaadni, ilyenkor ezek unióját közvetítjük a felhasználó számára. Belső tesztjeink

2 http://hunspell.sourceforge.net/

3 http://magyarispell.sourceforge.net/

alapján ismeretlen szóalakok esetén a konzervatívabb, de pontosabb Humor elemző eredményét vesszük alapul, így a szóalakot ismeretlennek tüntetjük fel akkor is, ha a Hunspell szerint ismert volt, de a Humor szerint nem.

9 Különírás-egybeírás

A Különírás-egybeírás modul (*Külön vagy egybe?*) ellenőrzi a megadott – akár helytelenül írt – (összetett) szót, vagy szavakat, illetve visszaadja a szabályok, illetve a rendelkezésre álló eszközök által biztosan megállapíthatóan helyesen (külön-, egybe-, illetve kis- vagy nagykötőjellel) írt alakokat, kiegészítve magyarázatokkal és hivatkozásokkal az AkH. megfelelő pontjaira.

A modul jelenleg nem fedi le az AkH. összes, különírással-egybeírással kapcsolatos rendelkezését. A szabályzat nyelvtechnológiai eszközökkel kezelhető területei közül jelenleg az alábbiakat valósítottuk meg:

- jelölt és jelöletlen alárendelői összetételek/szintagmák,
- a szótagszámlálási (6:3-as) szabály,
- mozgószabályok,
- rövidítéseket és mozaikszókat tartalmazó összetételek,
- néhány speciálisabb szabály, például a színnévi összetételek, anyagnévi összetételek stb.

A modul csak részben képes kezelni a jelentéssűrítő, illetve a szervetlen szóösszetételeket azok algoritmizálhatatlansága miatt. A mellérendelő (ikerszók, álikerszók stb.), valamint a morfológiai típusú összetételekkel foglalkozó szabálypontok – hasonló okokból – sincsenek beépítve.

A különírás-egybeírás helyesírási szabályrendszere felfogásunkban modellezhető egy generatív nyelvtani rendszerrel, hiszen a szabályzat rendelkezik arról, hogy milyen szóelemek milyen feltételek mellett, milyen írásmóddal (egybe-, külön, kötőjellel írás) vonhatók össze összetett szavakká, illetve szószerkezetekké (frázisokká). A szabályok egy része alkalmazható rekurzívan, illetve bizonyos szabályok láncokban egymásra is épülhetnek, így kézenfekvő egy formális nyelvtanra leképezni őket. Minden lehetséges nyelvtani levezetés (elemzési fa) megfeleltethető egy-egy értelmezésnek, illetve helyes írásmódnak. Attribútumok és értékadások segítségével az elemzési fák kiszámíthatják a bemeneti szóelemek közötti elválasztó karaktereket is (üres sztring – egybeírás esetén –, szóköz, kötőjel vagy nagykötőjel), melyekkel megadható, hogyan kell az adott értelmezés szerint helyesen leírni az összetett szót vagy kifejezést. Ha a fák felépítésekor feljegyezzük azt is, hogy egy adott összevonás (új csomópont létrehozása) melyik újraíró szabály alkalmazásával jött létre, a kész fa bottom-up bejárásával, a szabályokhoz rendelt magyarázó szövegek felhasználásával részletes segítséget tudunk generálni a felhasználó számára arról is, hogy milyen helyesírási szabályok alkalmazásával jött ki az adott megoldás és milyen értelmezési feltételek kapcsolódnak hozzá.

Célunk a fentieknek megfelelő formális nyelvtan kidolgozása, illetve egy, az ennek megfelelő kifejezéseket elfogadó, belső szerkezetüket feltáró nyelvtani elemző (parser) kifejlesztése volt. A környezetfüggetlen kifejezésnyelvtan újraíró szabályai a

morfológiai elemző által megadott szófaji, alaktani, szótagszámot, szóelemek számát tartalmazó, illetve az adatbázisainkból rendelkezésre álló szemantikai tulajdonságokból generált attribútum-érték szerkezetekkel operálnak. Utóbbiak biztosítják, hogy a szavak különböző fogalmi csoportjaira (anyagnevek, színnevek stb.) építő helyesírási szabályokat alkalmazni tudjuk.

Az AkH. a szabályokhoz illeszkedő analogikus alakok mellett nagymértékben tartalmaz kivételeket is. Igyekeztünk az összes, általunk implementált helyesírási részterületben megtalálható kivételes esetet azonosítani és felvenni őket a lexikai adatbázisba. A kivételek ellenőrzéséről és kezeléséről külön mechanizmusok gondoskodnak (l. később).

A következő részekben részletesen ismertetjük azokat a lépéseket, amelyek a felhasználói bemenetből a nyelvtani elemző számára feldolgozható elemeket állítanak elő, meghívják a kifejezésnyelvtanra épülő elemzőt, végül az elemzési fákból generálják a helyes írásmódokat és a hozzájuk tartozó magyarázó szövegeket. A modul futtatásának fő lépései vázlatosan a következők:

1. Előfeldolgozás:
   - A felhasználói bemenet egyszerű ellenőrzése,
   - A bemenet szegmentálása elemi tokenekre,
   - A tokenlista különböző írásmódjainak ellenőrzése kivételszótárainkban,
   - A tokenek felcímkézése morfológiai és szemantikai tulajdonságaikkal.
2. Elemzés: elemzési fák előállítása a kifejezésnyelvtan és az elemző (parser) segítségével
3. Kimenet: az elemzési fákból lehetséges helyes írásmódok, természetes nyelvű magyarázó szövegek generálása a felhasználó számára.

9.1 Előfeldolgozás

A modul számára érkező felhasználói bemenetet néhány egyszerű ellenőrzés után (karakterszám, ismétlődő karakterek ellenőrzése stb.) megkíséreljük tokenekre felbontani.

A felhasználó által megadott (normatív szempontból felesleges) kötőjeleket és egyéb írásjeleket szóközökre cseréljük, majd az így kapott elemeket megpróbáljuk további összetételi tagokra bontani. Ezek lehetnek akár önmagukban helyes, de a többi szó kontextusában, más szabályok szerint akár más módon (külön, kötőjellel stb.) is írható szóösszetételek tagjai (pl. „almafa” vö. „birsalma-fa”), akár helytelenül egy szóba írt kifejezések elemei is (pl. „pirosalma”). Az összetételi szabályokat ellenőrző kifejezésnyelvtan szabályait ezekre az atomi szinten szétválasztott tokenekre írtuk. Ehhez a művelethez a Humor morfológiai elemző egy speciális üzemmódját használjuk, amely képes a helyesírási szabályokkal nem konform összetett alakokhoz is elemzéseket előállítani.

Amennyiben nem tudtunk minden tokent a morfológiai elemzővel azonosítani, illetve tovább bontani, jelezzük ezt a felhasználó számára, és megkérjük, hogy próbálja meg kérdését, amennyire csak lehetséges, szavakra tagolva megismételni. Ha ez a bemenet sem értelmezhető, a folyamat hibaüzenet jelzésével véget ér, illetve az oldal átirányítja a felhasználót a *Helyes-e így*? és a *Névkereső* modulokhoz.

Sikeresen felbontott és morfológiailag elemzett bemenet esetén először ellenőrizzük, hogy a tokenek valamilyen lehetséges írásmódja nem szerepel-e kivétellistáink egyikében. N darab tokenből álló input esetén összesen $k^{(n-1)}$ írásmód lehetséges, ahol k a két szomszédos token közötti lehetséges elválasztó szimbólumok száma: k = |{egybeírás, szóköz, kötőjel, nagykötőjel}|. Az összes bemeneti tokent tartalmazó, általunk ismert kivétel azonosítása esetén a folyamat – a kivételes írásmód és megfelelő magyarázat jelzésével – véget ér.

Ha a bemeneti tokenek nem képeznek kivételt, sor kerülhet szószerkezet-nyelvtani elemzésük előkészítésére. Ehhez a már azonosított szófaji, morfológiai, szótagszámmal és összetételi tagok számával kapcsolatos információkon túl az adatbázis segítségével kikeressük a tokenekhez rendelhető szemantikai kategóriákat is.

Az egyes szabályok működését támogató adatbázis részét képezik a színnevek, foglalkozások és rangok, számnevek, földrajzi jellegű jelzők és köznevek, közterületek nevei, keresztnevek, népek és nyelvek nevei, rövidítések, közszói betűszavak, elő- és utótagok, a helyesírási szabályzatban az egyes szabályokban hivatkozott további kategóriák és különösen az egyes kivételek listája, mely jelenleg több mint 2100 szóból áll. Ezek a speciális tulajdonságok egy-egy szemantikai, illetve grammatikai jegyként vannak tárolva a szóalakok metaadatai között, és a szabályok számára közvetlenül hozzáférhetők.

9.2 A nyelvtan

A modul alapját képező környezetfüggetlen, jegystruktúrás nyelvtan formális leírása független a modul programkódjától, így könnyen karbantartható, fejleszthető. A nyelvtan a – jelenleg mintegy 160db – újraíró szabály megadásán kívül az alábbi elemeket tartalmazza:

- a szabály egyedi azonosítóját,
- a szabály alkalmazásának magyarázatát a felhasználó számára,
- hivatkozást az AkH. megfelelő szabálypontjaira és/vagy az Osiris-helyesírás [2] releváns témaköreire,
- példákat a szabály alkalmazására (az automatizált teszteléshez),
- valamint azokat a szabályokat, amelyek a szabályalkalmazó algoritmus futtatásakor konkurensek lehetnek az adott szabályra nézve; ilyenkor ezeket a szabályokat letiltjuk az alkalmazását.

Az újraíró szabályok a következőképpen néznek ki: *X(a=v, ...) + ... == Y(a=v, ...)*, ahol *X* bal oldali szimbólum, *Y* jobb oldali szimbólum, *a* egy attribútum neve, *v* ennek értéke. (A szabályokban, a konvenciótól eltérően a bal oldalon szerepelnek azok a szimbólumok, amelyekből egy elemzési lépésben összevonást végzünk a jobb oldalon megadott szimbólumba.) A bal oldali szimbólumokban az attribútum-érték párok az inputra érvényes megszorításokat, a jobb oldali szimbólumokban értékadásokat jelentenek.

A leíró nyelvtan szimbólumai megállapodás szerint az angol szófaji kategóriák kezdőbetűi vagy -betűcsoportjai: N (főnév), A (melléknév), V (ige), Adv (határozószó), Num (számnév). A szabályok bal oldalán a következő attribútumok állhatnak:

- Szemantikai jegyek listája (**sem**); a külön- vagy egybeírás kérdése (a szerkezetet alkotó szavak összetételi tagjainak számán kívül) bizonyos esetekben ezen dől el, pl.: *arany + gyűrű = aranygyűrű* (egybe), *fehérarany + gyűrű = fehérarany gyűrű* (külön) stb. Ebben az esetben feltétlenül szükséges az a többlettudás az *arany* szóról, hogy anyagnévről van szó.
- A **match** attribútum értéke egy reguláris kifejezés, amely illeszkedik a morfológiai elemző által előállított címkesorozatra. Például az alanyos vagy tárgyas viszonyt kifejező birtokos jelzői alárendelések (genitivus obiectivus/subiectivus) esetében a második tag mindig egy -ás/-és képzős ige: *match= "IGE,_IK,NOM"*, ahol *_IK* az -ás/-és képzőt jelöli.
- A bemenet felszíni alakja (**wordform**), illetve annak töve (**stem**).
- Az **ncomparts** attribútum azt mondja meg, hogy pontosan hány összetételi tagból áll az adott szimbólumnak megfelelő token-(rész)sorozat, az **ncompartsx** ennek alulról korlátos megfelelője.
- Az **nsylls** attribútum az adott szó szótagjainak számát adja meg (erre a szótagszámlálási szabálynak [más néven 6:3-as szabálynak] van szüksége).
- A **join1**, **join2** attribútumok a kivételes (nem formalizálható) írásmódú összetételek kezelésére szolgálnak. Az előfeldolgozás során, ha a tokenek felszíni alakjai valamilyen kombinációban szerepeltek a kivételszótárban, megkapják értékül a kivétel kategóriáját (pl. *Jelentessurito*), így az adott kivételeket kezelő szabályok érvényesek lesznek rájuk.

A jobb oldali értékadásban csak a tokenek közé kerülő elválasztó jeleket kódoló *sep* attribútum, illetve az összetett alak tagjainak számát megadó *ncompartsx* attribútum kerül.

A fej (a jobb oldalon az utolsó szimbólum) bizonyos jegyeit automatikusan megörökli a szabály jobb oldalán álló szimbólum, ha értékük specifikálva van (pl. *sem* attribútum: ha a fej például egy színnév, akkor a szabály által generált szimbólum is egy színnév lesz.)

9.3 A parser

A parser egy hagyományos bottom-up modellt valósít meg, a terminálisok többféle lehetséges értelmezéséből eredő összes értelmes feldolgozási fáját előállítja. A terminálisok többértelműségét az algoritmus csak akkor oldja fel, ha az adott értelmezés a szabályok megfelelő alkalmazási sorrendje mellett teljes fává összeáll. Ez a többértelműség korai feloldásánál nagyobb számítási igénnyel jár ugyan, de pontosabb (lásd pl. [9]), valamint sztochasztikus tényezőktől mentesen garantálja, hogy a szabályok megfelelő fedése esetén a végeredményül kapott fák halmaza tartalmazza a helyes eredményt is.

A terminálisok többértelműségén túl az alkalmazható szabályok halmaza és sorrendje sem egyértelmű. Bottom-up megközelítésről lévén szó, a helyes sorrend legenerálását csak kipróbálás útján lehet megtalálni.

Formálisan:

1. Legyen $F_1..F_k$ k db izolált fa, azaz k-komponensű erdő, amelyek levélpontjainak halmaza pontosan egybeesik a terminálisok halmazával, úgy, hogy az egyes fák levelei összefüggő szövegrészeket fednek le, és az F_i-k indexelési sorrendje egybeesik a terminálisokéval. A lehetséges szabályalkalmazások helye ezen fák $V_1..V_k$ gyökérpontjain lesznek.

2. Legyen H a lehetséges szabályalkalmazások halmaza, ahol H egy tetszőleges m-argumentumú h elemére jellemző, hogy az $V_a..V_{a+m-1}$ csomópontokra illeszkedik, és létrehoz feléjük egy új, G_h csomópontot. Ez m-1-gyel csökkenti a komponensek számát. Az algoritmus H minden elemére lemásolja az F_i erdőt, és a másolatokon sorra alkalmazza H szabályait.

3. Ha H üres volt, az erdőn nincs szabályalkalmazás, ami azt jelenti, hogy a rendszer szabályai szerint a terminálisok a jelen behelyettesítési értékük mellett nem állnak össze.

4. Ha az erdő egyetlen fává összeállt, az jó megoldás.

5. Amennyiben a szabályalkalmazás hatására még nem állt össze fává az erdő, a jelenlegi állapottal újra elindul az 1. lépéstől.

Az algoritmus futási ideje a terminálisok lehetséges értelmezéseinek számától ($t_1..t_n$), valamint az egy csomópontsorozatra illeszkedő szabályok maximális számától (m) függ. Felső becslés a legrosszabb esetű lépésszámra:

$$L = O(m)O(n^2)O(\prod_{i=1}^{n} t_i) = O(mn^2t^n)$$

ahol t a t_i-k maximuma. Ez n-ben hiperexponenciális, de csak irreális feltételek mellett valósulhat meg. A fenti implementációval az átlagos eset n-ben még mindig exponenciális, de nagy bemenet mellett található olyan heurisztika, amely segítségével a tényleges lépésszám n-ben csak polinom lesz.

9.4 Kimenet

A kifejezésnyelvtan és az elemző segítségével előállított elemzési fák tartalmaznak minden olyan információt, melyek segítségével a bemenethez megadható összes lehetséges helyes írásmód előállítható és ezekhez megfelelő magyarázatok fűzhetők. Az alábbiakban egy példán keresztül szemléltetjük a feldolgozás egyes lépéseit és a felhasználó számára megadott kimenetet.

A felhasználói bemenet legyen az alábbi:

```
sötétnarancssárga
```

Az előfeldolgozás során ezt a következő tokenekre választjuk szét:

```
sötét narancs sárga
```

A tokenekből a morfológiai elemző és a szemantikai jegyek adatbázisa segítségével az alábbi terminális szimbólumokat és attribútum-érték struktrúrákat állítjuk elő:

```
1.
N(wordform="sötét", stem="sötét", match="FN,NOM", sem=['Color3'],
ncomparts="1", ncompartsx="1+", nsylls="2")
A(wordform="sötét", stem="sötét", match="MN,NOM", sem=['Color3'],
ncomparts="1", ncompartsx="1+", nsylls="2")
```

```
2.
N(wordform="narancs", stem="narancs", match="FN,NOM", sem=['Color1'],
ncomparts="1", ncompartsx="1+", nsylls="2", join1=['Color1'])

3.
A(wordform="sárga", stem="sárga", match="MN,NOM", sem=['Color1'],
ncomparts="1", ncompartsx="1+", nsylls="2", join2=['Color1'])
```

Látható, hogy az első tokennek két különböző szófajú elemzése is van (főnév, melléknév). A `Color1` és `Color2` szemantikai jegyek a színnevek, illetve a színárnyalatok kategóriáit jelentik.

A parser segítségével a terminális struktúrákból első lépésben az alábbi elemzési fák építhetők:

```
1.
A(sep=[' '], ncompartsx="2+", sem=['Color1']) : M_EK_SZIN_3
    A(stem="sötét", sem=['Color3'])
    A(sep=[''], ncompartsx="2+", sem=['Color1']) : M_EK_SZIN_1_2
        N(stem="narancs", sem=['Color1'])
        A(stem="sárga", sem=['Color1'])

2.
A(sep=[' '], ncompartsx="2+", sem=['Color1']) : M_EK_MINOSEG_1_2
    A(stem="sötét", sem=['Color3'])
    A(sep=[''], ncompartsx="2+", sem=['Color1']) : M_EK_SZIN_1_2
        N(stem="narancs", sem=['Color1']
        A(stem="sárga", sem=['Color1'])
```

A fenti (egyszerűsített, nem az összes attribútumot megjelenítő) fákban a nemterminális szimbólumok után, kettősponttal elválasztva az őket létrehozó szabályok azonosítója olvasható. Az eltérés a fenti – helyesírás szempontjából egyforma végeredményt adó – két elemzés között az, hogy az `M_EK_SZIN_1_2` szabály (*Összetett színnevek képzése*) alapján egybe írt *narancssárga* tokent a *sötét* jelzővel 2 különböző szabállyal is összevonhatjuk: `M_EK_SZIN_2` (*Színárnyalat és színnév összetétele*) és `M_EK_MINOSEG_1_2` (*Minőségjelzős szerkezetek*).

A többértelműségek csökkentése érdekében a nyelvtan szabályai között részleges rendezési relációt definiáltunk, ez a gyakorlatban egyes szabályok más szabályokra vonatkozó letiltásával valósul meg (l. 9.2 rész). Ebben az esetben a specifikusabb `M_EK_SZIN_2` szabály tartalmaz egy tiltó utasítást az általánosabb `M_EK_MINOSEG_1_2` szabályra nézve, ennek eredményeképpen a parser kimenetében csak az 1. elemzési fa fog megjelenni.

Az utolsó lépésben az elemzési fák alulról-felfelé bejárásával, a csomópontokhoz rendelt szabályazonosítók, a **sep** attribútum (a külön- vagy egybeírást kódoló szimbólumok) és a terminális kategóriák segítségével, előre megadott sablonok kitöltésével generáljuk a felhasználó számára az elemzéseknek megfelelő természetes nyelvű magyarázatokat. Az alábbiakban bemutatjuk az 1. elemzési fából generált kimenetet:

```
Javasolt alak: „sötét narancssárga”

Magyarázat:

  1.  A „narancs” főnév és a „sárga” melléknév az alábbi szabály alap-
      ján egybeírandó:
      Az összetett színneveket egybeírjuk.
```

```
2. A „sötét” melléknév és a „narancssárga” melléknév az alábbi sza-
   bály alapján különírandó:
   A színnévi alaptag összetett, ezért különírjuk jelzőjétől (AkH.
   110.)
```

10 Összefoglalás

A dolgozatban bemutattuk a *helyesírás.hu* helyesírási tanácsadó portál hátterét, nyelvtechnológiai támogatással működő webalkalmazásainak részletes működését. Természetesen csak a valós használat során lehet majd felmérni, hogy a jelen formájában már működőképes, de lehetséges problémáknak csak korlátozott körét kezelni képes rendszer milyen felhasználói elégedettségi mutatókra számíthat, mint ahogy a mindennapi használatban felmerülő kérdések határozzák meg azt is, hogy a további fejlesztéseknek milyen területekre kell fókuszálni.

Hivatkozások

1. Kis Á.: Az akadémiai helyesírási szabályzat és a számítógép. Magyar Nyelvőr, Vol. 123, No. 2 (1999)
2. Laczkó K., Mártonfi A.: Helyesírás. Osiris Kiadó, Budapest (2005)
3. Naszódi M.: Nyelvhelyesség-ellenőrzés számítógéppel (Parciális szintaxis). In: VII. Országos Alkalmazott Nyelvészeti Konferencia, I. kötet. Külkereskedelmi Főiskola, Budapest (1997) 256–260
4. Németh, L.: Automatic non-standard hyphenation in OpenOffice.org. TUGboat, Vol. 27, No. 2 (2006) 750–755
5. Novák A., M. Pintér T.: Milyen a még jobb Humor?. In: Alexin Z., Csendes D. (szerk.): A 4. Magyar Számítógépes Nyelvészeti Konferencia előadásai. Szegedi Tudományegyetem (2006) 60–69
6. Pintér T., Oravecz C., Mártonfi A.: Online helyesírási szótár és megvalósítási nehézségei. In: Tanács A., Szauter D., Vincze V. (szerk.): MSZNY 2009. Magyar Számítógépes Nyelvészeti Konferencia JATEPress, Szeged (2009) 172–182
7. Pomázi Gy. (szerk.): A Magyar Helyesírás Szabályai. Tizenegyedik kiadás, tizenkettedik (példaanyagában átdolgozott) lenyomat. Akadémiai Kiadó, Budapest (2009)
8. Prószéky, G., Kis, B.: Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inectional Languages. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. College Park, Maryland, USA (1999) 261–268
9. Yoshida, K., Tsuruoka, Y., Miyao, Y.: Ambiguous Part-of-Speech Tagging for Improving Accuracy and Domain Portability of Syntactic Parsers. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (2007)
10. Varasdi K., Rebrus P.: A helyesírás mint default öröklődési hálózat. Előadás A mai magyar nyelv leírásának újabb módszerei VI. konferencián. Szeged (2003)

Helyesírási hibák automatikus javítása orvosi szövegekben a szövegkörnyezet figyelembevételével

Siklósi Borbála[1], Novák Attila[1,2], Prószéky Gábor[1,2]

[1] Pázmány Péter Katolikus Egyetem Információs Technológiai Kar,
[2] MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter u. 50/a
{siklosi.borbala, novak.attila, proszeky}@itk.ppke.hu

Kivonat: Cikkünkben egy korábban bemutatott orvosi helyesírás-javító rendszer lényegesen továbbfejlesztett változatát mutatjuk be, amely a korábbival ellentétben képes az egybeírások javítására, és a szövegkörnyezetet is figyelembe veszi ennek során, így alkalmas teljesen automatikus javításra is.

1 Bevezetés

A kórházakban keletkező szöveges dokumentumok olyan mennyiségű és minőségű gyakorlati tudást tartalmaznak, melyek feldolgozása és az eredmények felhasználása társadalmi szempontból hasznos, hozzájárulhat a ma sokszor hangsúlyozott életminőség javításához is. Mivel ezek a szövegek egyrészt mindenféle kontroll (pl. helyesírás-ellenőrző) alkalmazása nélkül készültek, másrészt az adott szövegtípusban nagyon magas arányban fordulnak elő a köznapi nyelvhasználattól idegen szóalakok: latin szavak, rengeteg rövidítés, gyógyszernevek, amelyeknek a helyesírására vonatkozó normákkal a szövegek íróinak nagy része nyilvánvalóan nincs tisztában, az ilyen szövegeknek a javítása nem könnyű feladat. A megvizsgált klinikai szövegekben jellemzően jelen vannak a hivatalos normától eltérő használatból fakadó, de következetesen elkövetett hibák, a véletlen melléütések, a következetlen szóhasználat, illetve az olyan többértelmű elírások, melyek helyességének megítélése még orvosi szakértelemmel sem egyértelmű (pl. elírt rövidítések). Emellett jellemző még az általános helyesírás-ellenőrzés során is felmerülő további probléma is: önmagukban helyes, de az adott környezetben téves szóalakok is előfordulnak. Mivel a szövegekben sok kifejezés gyakran egyáltalán nem fordul elő a helyesírási normának megfelelő formában, ezért úgy találtuk, hogy a gyakorlatban nem érdemes a szövegeket a helyesírási normáknak tökéletesen megfelelő formára hozni, ehelyett elegendő a szövegek egységesítése.

Cikkünkben bemutatjuk, hogy egy korábban létrehozott, helyesírási hibákat felismerő, s azok javítására automatikus javaslatokat generáló rendszer továbbfejlesztése során milyen stratégiákat alkalmaztunk a szövegkörnyezet figyelembevételére, illetve a generált javaslatok közül a megfelelő jelölt kiválasztására. Bemutatjuk, hogy az így létrehozott rendszer pontossága jelentősen javult, illetve az algoritmus kifinomultságának köszönhetően a paraméterek módosításával érzékenyebben hangolható. Ez a

megoldás könnyen kiterjeszthető azoknak a szavaknak a kezelésére is, amelyekben egynél több hiba van, mellyel további javulást érhetünk el. Így egy olyan rendszer kifejlesztéséről számolunk be, amely a jelentős minőségbeli javulás mellett használhatóságában is közelebb került egy teljesen automatikusan működő eszköz megalkotásához, mellyel az orvosi szövegek normalizálása megoldhatóvá válik a további felhasználási lépések előkészítéseként.

2 Helyesírási hibák

A klinikai dokumentumok jellegzetessége, hogy gyorsan, utólagos lektorálás, ellenőrzés, illetve automatikus segédeszközök (pl. helyesírás-ellenőrző) nélkül készülnek, ezért a leírás során keletkezett hibák száma igen nagy, valamint sokféle lehet. Így nem csupán a magyar nyelv nehézségeiből eredő problémák jelennek meg, hanem sok olyan hiba is felmerült a szövegekben, melyek a szakterület sajátosságaiból erednek. A legjellemzőbb hibák az alábbiak:

- elgépelés, félreütés, betűcserék,
- központozás hiányosságai (pl mondathatárok jelöletlensége) és rossz használata (pl. betűközök elhagyása az írásjelek körül, illetve a szavak között),
- nyelvtani hibák,
- mondattöredékek,
- a szakkifejezések latin és magyar helyesírással is, de gyakran a kettő valamilyen keverékeként fordulnak elő a szövegekben (pl. tensio/tenzio/tensió/ tenzió); külön nehézséget jelent, hogy bár ezeknek a szavaknak a helyesírása szabályozott, az orvosi szokások rendkívül változatosak, és időnként még a szakértőknek is problémát jelent az ilyen szavak helyességének megítélése,
- szakterületre jellemző és sokszor teljesen ad hoc rövidítések, amelyeknek nagy része nem felel meg a rövidítések írására vonatkozó helyesírási és központozási szabályoknak.

A fenti hibajelenségek mindegyikére jellemző továbbá, hogy orvosonként, vagy akár a szövegeket lejegyző asszisztensenként is változóak a jellemző hibák. Így elképzelhető olyan helyzet, hogy egy adott szót az egyik dokumentum esetén javítani kell annak hibás volta miatt, egy másik dokumentumban azonban ugyanaz a szóalak egy sajátos rövidítés, melynek értelmezése nem egyezik meg a csupán elírt szó javításával.

A feladat másik nehézségét az jelentette, hogy egyáltalán nem állt rendelkezésünkre nagyméretű helyesen írt klinikai korpusz, amely alapján elő tudtunk volna állítani a javításhoz használható hibamodelleket.

3 Statisztikai gépifordító-rendszer helyesírási hibák javítására

Célunk egy korábban [13]-ban már bemutatott, csak izolált szavak alapvető tulajdonságait figyelembe vevő és ezeket alkalmazó rendszer továbbfejlesztése volt. A fent leírt nehézségek miatt a rendszer elsősorban a szakterület nyelvére épített statisztiká-

kat vette legnagyobb súllyal figyelembe – természetesen a morfológia mellett –, ami biztosítja a speciális szövegek sajátosságainak megtartását az általános szövegekből átvett formák alkalmazása helyett. A korábbi eredmények során bemutattuk, hogy az így létrejött rendszer a szövegekben lévő hibákat felismeri, az azokhoz automatikusan generált és rangsorolt javaslatok között az első tíz az esetek 98%-ában tartalmazta a helyes alakot.

Mivel célunk a háttérben futó automatikus normalizálás, és nem az, hogy a javaslatokat egy felhasználónak mutassa meg a rendszer, aki aztán kiválasztja a helyes alakot, ezért fontos, hogy a rendszer képes legyen a generált javaslatok közül a valóban helyeset automatikusan kiválasztani. A legjobb javítás kiválasztásához kevésnek bizonyult a korábbi rendszerben alkalmazott, kizárólag morfológiára és különböző szóstatisztikákra épülő rangsorolás. Az automatikus javítás pontosságának növeléséhez szükséges az egyes szavakhoz tartozó szövegkörnyezet figyelembevétele is. E két követelmény alkalmazására a statisztikai gépi fordítás területén széles körben alkalmazott Moses keretrendszert használtuk. A fordítás során forrásnyelvnek az eredeti hibás szöveget tekintjük, míg a célnyelv ennek javított formája. Ebben az esetben a rendszer bemenete a hibás mondat: $H=h_1,h_2...h_n$, melynek megfelelő javított mondat a $J=j_1,j_2..j_k$ a várt kimenet. A helyesírás-javító rendszer zajos csatornamodellként tehát úgy fogalmazható meg, hogy az eredeti üzenet a helyes mondat, amely helyett a csatornán átért jel a zajos, azaz hibás mondat. Így a javítás az a $\hat{J}$ mondat lesz, melyre a

$$\hat{J} = argmax\ P(J|H) = argmax\ P(H|J)P(J) / P(H) \quad (1)$$

feltételes valószínűség a maximális. Mivel $P(H)$ értéke állandó, ezért a nevező elhagyható, így a számlálóban lévő szorzat a fordítási és nyelvmodellből számított statisztika alapján számítható.

Ezeket a modelleket a hagyományos statisztikai gépifordító-rendszerek esetén a forrás- és célnyelvű párhuzamos korpuszból számolt valószínűségek képezik. Ilyen korpusz azonban a mi esetünkben nem áll rendelkezésre, ezért a fordítási modellt a korábban létrehozott rendszer rangsorolásához használt számítási értékek valószínűségekké konvertálása képezi. A szövegkörnyezet figyelembevétele érdekében pedig a SRILM eszköz segítségével létrehozott nyelvmodell módosítja a "fordítás" során kapott eredményeket.

3.1 A fordítási modellek

A rendszerben három szó-, illetve hibatípus kezelésére három fordítási (hibajavítási) modellt alkalmazunk. Az egyik kifejezetten a rövidítések kezelésével, a másik a tévesen egybeírt szavak felbontásával, a harmadik pedig az egyéb elírásokkal foglalkozik. A modellek bemutatását ez utóbbival kezdjük.

3.1.1 Fordítási modell általános szavakra

A fordítási modellt, vagy más néven frázistáblát a korábban implementált javaslatgeneráló eredménye alapján építettük. Minden javítandó szóhoz (a potenciális rövidíté-

seket és stopszavakat külön kezeljük) az első 20-ként rangsorolt javaslatot vettük figyelembe. Ezek között a javaslatok között szerepelhet olyan eset is, amikor az eredeti alakba szóköz kerül, így az esetleg hibásan egybeírt szavak szétválasztásának lehetősége is belekerült a rendszerbe. Mivel korábban bemutattuk, hogy a javaslatok generálása során az esetek 98%-ában az első tíz javaslatban (amely még nem tartalmazott szóközbeszúrási lehetőséget) benne volt a helyes alak, ezért 20-nál több lehetőség figyelembevétele még a különírások figyelembevételével is csak fölösleges zajt generált volna. Az így megkapott javaslatok rangsorának kialakításához használt pontszámot (l. [13]) alakítottuk valószínűséggé, azaz az egy szóhoz tartozó lehetséges javítások valószínűségének összege 1. Ezzel a módszerrel helyettesítettük a párhuzamos korpuszból való tanítást.

1. táblázat: Részlet az általános szavakat tartalmazó frázistáblából.

hosszúságu ||| hosszúsági ||| 0.016497642667 ||| |||
hosszúságu ||| hosszúságú ||| 0.0156006851784 ||| |||
hosszúságu ||| hosszúsága ||| 0.013537671904 ||| |||
hosszúságu ||| hosszúságuk ||| 0.013537671904 ||| |||
hosszúságu ||| hosszúságul ||| 0.013537671904 ||| |||
hosszúságu ||| hosszúságé ||| 0.013537671904 ||| |||
hosszúságu ||| hosszúság ||| 0.013537671904 ||| |||

3.1.2 Fordítási modell rövidítésekre

A klinikai szövegekre jellemző, hogy az általános szövegeknél sokkal magasabb arányban tartalmaznak rövidítéseket. Ezek a fenti modellel két okból nehezen kezelhetőek. Egyrészt gyakran ugyanannak a szónak vagy kifejezésnek számos különböző rövidített alakja előfordul a szövegekben a dokumentum rögzítőjének egyedi szokásai miatt, vagy mert éppen úgy sikerült. Másrészt pedig ezekhez általában létezik több olyan generált javaslat, amelyek más, helyes szavakká alakítják ezeket, s mivel az ilyen valódi szavak előfordulása gyakoribb minden korpuszban, illetve a morfológia is megerősíti ezek helyességét, ezért könnyen átíródnának a szándékolt eredeti jelentéstől teljesen eltérő szóalakokká. Ezért automatikus módszerek alkalmazásával kigyűjtöttük az orvosi dokumentumokban előforduló rövidítéseket, melyek egy részét kézzel történő szűrés után a morfológiába is beépítettük. A fordítórendszer számára azonban ez az eljárás nem elégséges, hiszen egy-egy rövidítésnek számos alakja fordul elő a szövegekben (legjellemzőbb példa a rövidítések végén a pont elhagyása stb.). Ezért minden potenciális rövidítéshez kigyűjtöttük a dokumentumokból ezek variációit, gyakorisággal együtt, majd az így kapott gyakoriságokat szintén valószínűségekké alakítottuk. Így létrejött egy alternatív frázistábla a fordítórendszer számára. Mivel az ebben a modellben szereplő szóalakokhoz az előző pontban leírt módon nem generálunk javaslatokat, hogy ne javítsunk rövidítéseket egész más szavakká, ezért ezek csak ebben a második frázistáblában szerepelnek, az elsőben nem. A fordítórendszert úgy

alakítottuk ki, hogy az a fordítás során a bemenetre érkező szóalakhoz abból a táblából számít fordítási lehetőséget, amelyikben a szóalak megtalálható.

2. táblázat: Részlet a rövidítéseket tartalmazó frázistáblából.

conj. ||| conj. ||| 0.607803468208 ||| |||
conj ||| conj. ||| 0.869653179191 ||| |||
conj ||| conj ||| 0.130346820809 ||| |||
mko ||| mko ||| 0.489190805776 ||| |||
mko ||| mko. ||| 0.997094762027 ||| |||
mko. ||| mko. ||| 0.999316414595 ||| |||

3.1.3 A téves egybeírásokat kezelő modell

Mivel a gépi fordításra használt keretrendszert általában frázisalapú fordításra alkalmazzák a többnyelvű fordítás során, ezért általános jellemzője a hagyományos módon használt rendszerben lévő frázistáblának, hogy abban egy (vagy több) szóhoz tartozhat több szóból álló fordítás is. Így a mi esetünkben sem okozott problémát az, hogy nem csupán szóalapú megfeleltetéseink vannak, hanem egy szóból esetlegesen több szó is képződhet. Természetesen ezekhez is a javaslatgeneráló által kapott pontszámból számított valószínűség került a modellbe. Ezekben az esetekben a javaslatok pontszáma úgy adódik, hogy a szóköz beszúrásával kapott két lehetséges szóra számolja ki az értékeket, majd átlagolja, így kap a két szóból álló javaslat egy olyan pontszámot, amely nagyságrendileg illeszkedik az egy szóból álló javaslatok listájába.

3. táblázat: Részlet az általános szavakat tartalmazó frázistáblából.

soronkívül ||| soron kívül ||| 0.0207452583298 ||| |||
soronkívül ||| soronkívül ||| 0.0145949359186 ||| |||

3.2 A nyelvmodell

A nyelvmodell szerepe a javítás során az, hogy a fordítási modell alapján létrejött javított mondatokban szereplő szavak sorozatának előfordulási lehetőségét ellenőrizze, és a valós előfordulás felé súlyozza. Ennek a komponensnek a feladata, hogy a javítás során a szövegkörnyezetet is figyelembe vegye a rendszer. Ehhez az adott szövegtípusra jellemző helyes korpuszból kellene a kívánt hosszúságú szósorozatokat (szó n-eseket) tartalmazó statisztikát létrehozni. Mivel a rendelkezésünkre álló dokumentumoknak csak a tesztelésre használt része az, amely kézzel ellenőrzött módon helyesnek tekinthető, ezért nem volt lehetőségünk helyes korpuszból tanított nyelvmodellt létrehozni. Bár más témájú, illetve más stílusú szöveges korpuszok természetesen

léteznek, az ezekben található szó n-esek nem feltétlenül modellezik jól a klinikai szövegeket, ezért úgy döntöttünk, hogy nem használunk ilyen szövegeket. Azt láttuk azonban, hogy a klinikai dokumentumokban számos olyan szófordulat, szósorozat van, ami nagyon gyakori, összességében viszont kevés számú különböző szó n-es fordul elő, azaz a klinikai dokumentumok nyelvezete viszonylag korlátozottnak tekinthető ilyen szempontból.

4. táblázat: Általános magyar nyelvű és orvosi szövegekben előforduló különböző n-gramok száma (800000 mondatos korpuszban).

	Általános szöveg	Orvosi szöveg
1-gram	873951	275609
2-gram	4794135	1409290
3-gram	7886616	2440636

Ezért a hibás szót tartalmazó szósorozatok esetén ugyanezeknek a szósorozatoknak a helyes előfordulása gyakoribb, tehát a nyelvmodell-statisztikát a vártnál kisebb mértékben tekinthetjük torznak a hibás szavakat is tartalmazó korpuszból építve. Természetesen a kiértékelés során a mérésekhez használt teszthalmaz mondatait már a nyelvmodell építése előtt különválasztottuk a korpusztól, hiszen az ezekben megtalálható hibás szósorozatokra teljesen illeszkedő n-eseket találnánk a teljes mondatra, ami viszont már azzal járna, hogy a javítás helyett az eredeti szóalakok kapnának nagyobb súlyt.

A korpuszt alkotó dokumentumokat az előfeldolgozás során végzett tokenizálással egy időben a feltételezhető mondathatároknál mondatokra is bontottuk. Az így kapott mondatok átlagos hosszát (8,58 token/mondat) figyelembe vettük a nyelvmodell építése során, ezért a nyelvmodellt úgy hoztuk létre, hogy az abban szereplő szó n-esek maximális hossza három token. A rövid mondatok miatt nem várható el ennél hosszabb n-gramok esetén az illeszkedés. Ezt méréseink is megerősítették: nagyobb n esetén a végeredmény rosszabb lett.

Fontos megjegyezni még, hogy a nyelvmodell létrehozása előre megtörténik, ezért a dekódoláshoz szükséges idő az egyes mondatok esetén nem növeli számottevő mértékben a javításhoz szükséges időt.

3.3 Dekódolás

A fenti modellek alapján az (1) képlet alkalmazásával számított eredmény meghatározását a fordítórendszer magját képező dekódoló algoritmus végzi. Ehhez a Moses keretrendszert alkalmaztuk, amely a statisztikai gépi fordítás területén a legelterjedtebb eszköz. A dekódolás paramétereit a konfigurációs fájlban lehet beállítani. Így könnyen és gyorsan változtathatóak a rendszer paraméterei, ami igen rugalmassá teszi azt. A dekódolás során minden bemeneti mondathoz a fent részletezett modellek alapján mondatszintű fordítások (azaz a mi esetünkben javítások) jönnek létre, melyben a szószintű javítási lehetőségeket a frázistábla, a szövegkörnyezet figyelembevételét

pedig a nyelvmodell biztosítja. A dekódolás során a következő paramétereket használtuk:

- A frázistáblák súlyozása: mivel a frázistáblákban szereplő szavak halmazai diszjunktak, ezért ezek súlyának beállítása független egymástól. A szövegek javítása valójában inkább azok egységesítését jelenti, nem pedig a szigorú értelemben vett helyesírási normához való igazítását. Ezért a rövidítések sokféle megjelenési formája miatt ezeknél fontosabbnak láttuk az átírást egy meghatározott formára (amely általában a ponttal jelölt alak), így a rövidítésekhez tartozó fordítási modell nagyobb súlyt kapott.
- Nyelvmodell: trigram nyelvmodellt alkalmaztunk, azaz a nyelvmodellben szereplő szó n-esek hossza maximum 3. A dekódolás során a nyelvmodell a fordítási modellnél alacsonyabb súlyt kapott, hogy megakadályozzuk a hibás szövegekből készült nyelvmodellben előforduló hibás n-gramok előtérbe kerülését.
- Átrendezési korlát: a különböző nyelvek közötti fordítás során szükséges lehet a megfeleltetett szavak sorrendjének is a megváltoztatása, a helyesírás-javítás során azonban ezt nem engedhetjük meg, hiszen a módosítások csak szavakon belül, illetve szóközök beszúrásával történhetnek, a szavak sorrendjén a javítórendszer nem változtathat.
- Mondathossz-különbség: mivel a javított mondat hossza sem térhet el jelentősen az eredeti mondattól (ha minden szóba beszúrnánk egy szóközt, akkor érnénk el az elvi korlátot, de a valóságban ilyen nem fordulhat elő, mondatonként két egybeírási hiba volt a legtöbb, ami a teszthalmazban szerepelt), ezért nem szükséges a dekódolás során a mondathossz-eltérést külön súllyal büntetni.

4 Eredmények

A rendszer kiértékeléséhez szükséges volt az orvosi dokumentumokból létrehozott teszthalmaz kézzel való kijavítása, így létrejött egy 2000 mondatból álló, vegyes tartalmú (különböző klinikai osztályok anyagaiból származó) teszthalmaz. A nyelvmodell létrehozásához a fennmaradó 978000 mondatból álló korpuszt használtuk. Mindkét részhalmaz csak szabad szövegekből álló mondatokat tartalmaz, tehát az amúgy is szabványos BNO-kódokkal párosított betegségmegnevezéseket, kódokat, mérési és laboreredményeket nem tartalmazott a korpusz. Ennek ellenére számos olyan "mondat" került mind a tanítóanyagba, mind a teszthalmazba, amelyek nehezen értelmezhető, speciális tartalmú szavakat, rövidítéseket, gyakran rövidítéssorozatokat tartalmaztak. Ezek helyességének a megítélése külön feladat, amihez megfelelő szakterületi ismereteink hiánya miatt egy vagy több általunk helyesnek ítélt változatot fogalmaztunk meg elfogadható javításnak. Ráadásul az elvi helyesírási szabályoknak megfelelő formára való hozást el kellett vetnünk. Ennek egyik oka, hogy a gyakorlati alkalmazás során sok esetben a helyesírási szabályoknak ellentmondó írásváltozatok a korpuszban sokkal gyakoribbak voltak, mint a helyesírási normának megfelelő változat (amely sok esetben a korpuszban egyáltalán nem szerepelt). Úgy véljük, hogy a szövegekben szereplő fogalmak visszakereshetőségét az egységesítés abban az esetben is lehetővé

teszi, ha az alkalmazott egységes alak nem azonos valamely helyesírási norma által szentesített alakkal. Ezért a kézzel kijavított teszthalmaz létrehozásakor mindezen szempontokat figyelembe véve annak több változatát is elkészítettük, a lehetséges javításokkal.

A kiértékelés során ezen a teszthalmazon különböző metrikák szerint végeztünk méréseket. A gépi fordítás minőségére általánosan elterjedt mérőszám a Bleu érték meghatározása. Mivel a felfogásunk szerint a javítás folyamatát is egyfajta fordításként értelmezhetjük, ezért az egyik mérőszámként mi is ezt a metrikát használtuk. Ennek lényege, hogy a javítás eredményét a referenciafordításhoz hasonlítva a szavak sorrendjét is figyelembe vevő módosított pontosságértéket számol. Emellett a helyesírás-javítás hagyományos értelemben vett feladata során szokásos fedés, pontosság, F-mérték hármas mentén is vizsgáltuk a rendszer minőségét.

Mivel célunk a korábban létrehozott javító rendszer eredményeinek javítása volt, ezért azt tekintettük alaprendszernek. Ahogy az 5. táblázatból látszik, ez a korábbi rendszer megtalálta a legtöbb hibát (magas fedésérték), ám a szövegkörnyezetet figyelembe nem vevő pusztán rangsoroláson alapuló javítás pontossága rosszabb volt, ezért a kettő átlagából számított F-mérték is kisebb, csupán 72% volt.

5. táblázat: Az automatikus helyesírás-javító rendszerek minősége automatikus metrikák alapján.

	Pontosság	Fedés	F-mérték	Bleu
Alaprendszer	0,70	0,75	0,725	-
SMT	0,8814	0,8857	0,8826	0,8085

A statisztikai fordítórendszer alkalmazásával jelentősen javultak a mért eredmények, a legjobb paraméterbeállítás során 88%-os F-mértéket értünk el. Több olyan konfigurációval is elvégeztük a javítást, melynek eredményei valamivel rosszabb értéket produkáltak, de bizonyos jelenségek kezelésére mégis alkalmasabbak voltak.

6. táblázat: Eredetileg hibás mondatok és a hozzájuk tartozó automatikus javítás az alaprendszerrel, illetve a statisztikai módszer használatával (melynek eredménye megegyezik a referencia javítással).

Eredeti mondat:	*csppent előírés szerint ,*
Alaprendszer javítása:	*cseppent előír és szerint ,*
SMT javítás:	*cseppent előírás szerint ,*
Eredeti mondat:	*th : mko tovább 1 x duotrav 3 ü-1 rec , íb : 2 x azoipt 3 ü-1 rec*
Alaprendszer javítása:	*th : mko tovább 1 x duotrav 3 ü-1 sec , kb : 2 x azoipt 3 ü-1 sec*
SMT javítás:	*th. : mko tovább 1 x duotrav 3 ü-1 rec , kb : 2 x azopt 3 ü-1 rec*
Eredeti mondat:	*/alsó m?fogsor .*
Alaprendszer javítása:	*/alsó műfogsor .*
SMT javítás:	*alsó műfogsor .*
Eredeti mondat:	*vértelt nyállkahártyák , kp erezett conjuctiva , fehér sclera .*
Alaprendszer javítása:	*vértelt nyálkahártyák , kp erezett conjunctiva , fehér sclera .*
SMT javítás:	*vértelt nyálkahártyák , kp. erezett conjunctiva , fehér sclera .*

5 Nehéz esetek

Az automatikus javítás során több olyan jelenség van, amelyeket a javítónak nem sikerül kezelnie. Néhány példa:

- Vannak esetek, amikor a javító egy helyes szót, egy másik helyes szóra ír át, illetve egy elírt szót nem a megfelelő, ámde helyes szóalakra javít, melyek az adott mondatban is helytállóak. Ilyen esetek a nyelvmodellben való előfordulások miatt kerülnek előtérbe, a korábban említett nyelvmodellel kapcsolatos problémák miatt - különösen a rövid mondatoknál. További probléma, hogy két szó megváltoztatása esetén nem kap elég hangsúlyt az ezért járó büntetés, aminek erősítése viszont a ténylegesen szükséges javítások esélyét csökkentené.

  eredeti mondat: homályos látást panaszol .
  javított mondat: homályos látás panaszok .

  eredeti mondat: panasz nem volt .
  javított mondat: panasza nem volt .

  (A második példában a mondat átírása helyes változatot eredményez, de a referencia és a valós környezet nem tartaná szükségesnek az átírást.)

- Az egynél több hibát tartalmazó szavak javítása egyelőre nem lehetséges:
  eredeti mondat: gyógyógyszerei : ld lázlap
  javított mondat: gyógyógyszerei : ld lázlap

6 Összefoglalás

Cikkünkben bemutattuk, hogy a nagyon zajos, magyar nyelvű, de speciális nyelvezetű és rövidítésekkel teletűzdelt orvosi szövegekben lévő helyesírási hibák automatikus javítása a szövegkörnyezet figyelembevételével elég nagy pontossággal elvégezhető. A helyesírási normáknak megfelelő szaknyelvi korpusz hiányában a szövegek egységesítésének egyik lehetséges útja egyelőre az lehet, ha a normát nem úgy értelmezzük, hogy csak a szabványos helyesírásnak megfelelő alakokat fogadjuk el, hanem a korpuszban túlnyomó többségben szereplő alakokat is normalizálási célpontnak tekintjük. Így létrejöhet a szövegek egységes reprezentációja, ami a további feldolgozás szempontjából alapvető fontosságú.

Természetesen a feladatot még nem oldottuk meg tökéletesen, bemutattuk azokat a hibajelenségeket, amelyek kezelése még előttünk áll. További terveink között szerepel, hogy a rendszer részévé tegyük a PurePos szófaji egyértelműsítőt is, amely olyan kiegészítő információkat biztosítana, melyekre támaszkodva tovább javíthatnánk a rendszer teljesítményét a jelenleg még kétséges esetekben. Ezen túl további javulást remélünk a nyelvmodell építésére használt korpusznak az automatikus javítások utáni

iteratív újraépítésétől, ami a jelenleg igen zajos nyelvmodell helyességén javítana, így annak torzító hatását kiküszöbölné.

Köszönetnyilvánítás

Ez a munka részben a TÁMOP 4.2.1.B – 11/2/KMR-2011–0002 pályázat támogatásával készült.

Hivatkozások

1. Dustin, B.: Language Models for Spelling Correction CSE 256 (2004)
2. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (2000) 286–293
3. Contractor, D., Faruquie, T.A., Subramaniam, L.V.: Unsupervised cleansing of noisy text. In: Proceedings of the 23rd International Conference on Computational Linguistics (2010) 189–196
4. Heinze, D.T., Morsch, M.L., Holbrook, J.: Mining Free-Text Medical Records. A-Life Medical, Incorporated (2001) 254–258
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. Association for Computational Linguistics, Prague, Czech Republic (2007) 177–180
6. Mykowiecka, A., Marciniak, M.: Domain-driven automatic spelling correction for mammography reports. In: Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM'06. Advances in Soft Computing, Heidelberg (2006)
7. Ehsan, N., Faili, H.: Grammatical and Context-sensitive Error Correction Using a Statistical Machine Translation Framework. In: Software Practice and Experience (2011)
8. Novák A.: Milyen a jó Humor? In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). Szegedi Tudományegyetem (2003) 138–145
9. Orosz, Gy., Novák, A.: PurePos — an open source morphological disambiguator. In: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science (2012) 53–63
10. Patrick, J., Sabbagh, M., Jain, S., Zheng, H.: Spelling Correction in Clinical Notes with Emphasis on First Suggestion Accuracy. In: 2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining (2010) 2–8
11. Pirinen, T.A., Lindén, K.: Finite-State Spell-Checking with Weighted Language and Error Models – Building and Evaluating Spell-Checkers with Wikipedia as Corpus. In: SaLTMiL Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages, LREC (2010) 13–18
12. Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Inquiries into Words, Constraints and Contexts (2005) 150–157

13. Siklósi, B., Orosz, Gy., Novák A., Prószéky G.: Automatic structuring and correction suggestion system for Hungarian clinical records. In: Proceedings of the 8th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages (2012) 29–34
14. Stevenson, M., Guo, Y., Amri, A., Gaizauskas, R.: Disambiguation of biomedical abbreviations. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing (2009) 71

Magyar nyelvű klinikai rekordok morfológiai egyértelműsítése

Orosz György, Novák Attila, Prószéky Gábor

MTA-PPKE Magyar Nyelvtechnológiai kutatócsoport
Pázmány Péter Katolikus Egyetem, Információs Technológiai Kar
1083, Budapest, Práter utca 50/a
e-mail:{oroszgy, novak.attila, proszeky}@itk.ppke.hu

Kivonat Cikkünkben azokat az eljárásokat mutatjuk be, amelyekkel a meglévő PurePos szóalaktani egyértelműsítő rendszert, valamint az abban alkalmazott HuMor morfológiai elemzőt egy klinikai dokumentumokból álló orvosi korpusz elemzésére adaptáltunk. Ismertetjük a rendszer fejlesztéséhez szükséges teszthalmaz létrehozásának lépéseit, a fejlesztés alatt álló egyértelműsítő építőelemeit, és az azokon végzett első doménadaptációs eljárásokat. Részletesen leírjuk a felhasznált morfológiai elemző tőtárának bővítési lépéseit, az egyértelműsítőben a morfológiafejlesztés egyes megoldásai mellékhatásaként fellépő hibákat és az azokra adott megoldásokat. Végezetül megmutatjuk, hogy az így kapott eszközzel relatív 41,86%-kal sikerült csökkenteni a címkéző által vétett hibák számát, megvizsgáljuk a fennmaradó hibákat, s javaslatokat teszünk azok javítására.

1. Bevezetés

A legtöbb kórházban az orvosi feljegyzések tárolása csupán archiválás és az egyes esetek dokumentálása céljából történik. Ezen adatok felhasználási lehetősége így csupán az egyes kórtörténetek manuális visszakeresésére korlátozódik. Korábban bemutattunk [1,2] egy olyan automatikus eljárást, amely az orvosi (azon belül is a szemészeti) rekordok helytelen szavait nagy százalékban javítani tudja. Ezen előfeldolgozási lépés után a mélyebb szemantikai összefüggések automatikus kinyeréséhez szükséges a dokumentumok mondatainak (morfo-)szintaktikai annotálása is.

A szófaji és ezzel együtt a morfológiai egyértelműsítés a nyelvtechnológia egyik alapfeladata, mely a hagyományos szövegfeldolgozási lánc elején áll. Eredményének használatához – az egészségügy esetén pedig még inkább – annak nagy fokú pontossága szükséges. Angol nyelvterületen számos alkalommal vizsgálták már statisztikai tanuló algoritmusok orvosi doménre való adaptálását, míg a magyar nyelvű klinikai dokumentumok ilyen típusú feldolgozására nem ismerünk hasonló eredményeket.

Kutatásunkhoz szükség volt egy manuálisan annotált kis méretű korpusz létrehozására – immár nem csak szemészeti típusú klinikai dokumentumokat feldolgozva – melyet a bemutatott egyértelműsítő módszerek finomhangolására, tesztelésére és mérésre használtunk. Az ellenőrzött és javított morfológiailag címkézett szöveg elkészítéséhez a rekordokat automatikusan főbb alkotórészekre bontottuk, melyekből a kinyert szöveges bekezdésekhez adaptáltuk a központozási hibákat javító és tokenizáló rendszert, a morfológiai elemzőt és az egyértelműsítő rendszert.

Írásunkban a fenti lépéseken túl ismertetjük a HuMor morfológiai elemző [3,4] adaptálása során alkalmazott eszközöket, eljárásokat. Bemutatjuk az egyértelműsítő rendszer orvosi doménre történő alkalmazása során felmerült tipikus hibaeseteket és az erre adott megoldásokat. Végezetül áttekintjük az így kapott rendszer és részeinek eredményességét.

2. A tesztkorpusz létrehozása

A [2] cikkben korábban ismertetett helyesírásilag korrigált tesztkorpusz morfológiai egyértelműsítő fejlesztéséhez nehezen alkalmazható. Ez az anyag elsősorban szószintű problémák vizsgálatára lett létrehozva, és csak egy nagyon szűk domén kis méretű korpuszából vett szókincsével rendelkezik. Jelen kutatásunk keretében a korábbinál szélesebb domént lefedő és nagyobb terjedelmű tesztkorpuszt hoztunk létre. Az újonnan előállított korpusznak az alábbi feldolgozási lépéseken kellett keresztülmennie: a dokumentumok önálló strukturális egységekre tagolása, a központozási hibák automatikus javítása, mondatokra bontás és tokenizálás, a helyesírási hibák javítása, a szöveg automatikus morfológiai annotálása és annak manuális ellenőrzése, javítása. A munkánk során célunk olyan algoritmusok, módszerek készítése volt, mely segíti, támogatja morfoszintaktikailag egyértelműsített korpusz előállítását.

A korábbi XML-struktúrát létrehozó szabályalapú rendszer nem volt alkalmazható a szemészeti doménen kívül, mivel a a dokumentumok struktúrája osztályonként és akár orvosonként más és más. Így a szemantikai egységek meghatározásakor úgy döntöttünk, hogy a bekezdéseket tekintjük önálló összetartozó egységeknek. A bekezdésekre bontást egy, a formai jellemzők alapján működő egyszerű szabályalapú rendszer végezte, mely már általánosan alkalmazható volt. A bekezdéseket a további feldolgozás érdekében két osztályba kellett sorolni: főként nyelvi szöveget tartalmazó és egyéb, nem szöveges adatot tartalmazó bekezdésekre. Az osztályozáshoz az alábbi jellemzőket nyertük ki az egyes szakaszokból: sorok hossza, átlagos sorhossz, a legrövidebb sor hossza, átlagos soronkénti szószám, átlagos szóhossz, szavak száma, leghosszabb szó hossza, (feltételezhető) orvosnevek száma, egy szóból áll-e a bekezdés, whitespace karakterek aránya, írásjelek aránya, nagybetűk aránya, számszerű tokenek aránya, alfanumerikus karakterek aránya. Bár végeredményként a dokumentumok két osztályát kívántuk látni, azt tapasztaltuk, hogy a rendelkezésre álló adatokon ez a legtöbb közkedvelt gépi tanulásos algoritmusnak csak alacsony

eredményességgel sikerül, így a szövegek struktúrájához jobban illeszkedő alábbi osztályozást választottuk:

1. szöveges bekezdések,
2. fejlécek, szakaszcímek,
3. numerikus, illetve táblázatos adatok.

Egy kézzel ellenőrzött 500 bekezdésből álló teszthalmazon a klasszifikációs feladatra a J48 [5,6] döntési fa algoritmus bizonyult a legeredményesebbnek 93,2%-os keresztvalidált pontossággal.

Mielőtt a szövegeket a Huntoken rendszerrel [7] tokenizáltuk volna, az alábbi központozási hibákat javítottuk:

– a mennyiség és a mértékegység egybeírása,
– dátumok tagolatlansága,
– számszerű kifejezések egybeírása,
– jobbról tapadó írásjel (pont, vessző stb.) következő tokenhez való tapadása,
– központozási jeleknél whitespace-ek hiánya.

A szövegeinkben gyakori jelenség volt még a mondatvégi írásjelek hiánya, így a mondatokra bontás hibáját minimalizálva az egyértelmű helyeken tovább daraboltuk a bekezdést, így elkerülve, hogy több mondatot összevonva hibás határok kerüljenek megállapításra. (Pl. olyan sorok, amelyek csak rövid szöveget tartalmaznak a sor elején, nem vonandóak össze a következővel.) A mondatokra bontó alrendszerhez szükség volt még egy rövidítéslistára is, mely olyan – formai jegyeknek megfelelő – gyakori szavakból áll, melyeket automatikus módszerekkel illetve manuálisan is ellenőriztünk. (Pl.: a pont nélkül a HuMor által helyes szónak talált szóalakokat külön ellenőriztük.)

A véletlenszerűen választott 600 mondatból álló tesztanyag helyesírását, a már ismertetett [2] rendszerrel automatikusan javíttattuk, majd kézzel ellenőriztük és tovább javítottuk, majd a bemutatott egyértelműsítő alaprendszer kimenetét használva manuálisan annotáltuk a korpuszt.

3. Az egyértelműsítő rendszer kialakítása

3.1. A PurePos rendszer

Korábban ismertettük a PurePos [8] morfológiai egyértelműsítő rendszert, mely hatékonyan képes szófaji egyértelműsítésre és lemmák automatikus meghatározására. Bemutattuk, hogy a készített rendszer mind sebességben, mind pedig teljesítményben felveszi a versenyt társaival. A Szeged Korpuszon [9] tanítva és mérve 98,35%-os teljes pontosságról számolhattunk be. Integrált módon képes morfológiai elemzőt használni, mely a címkézés pontosságát – kis méretű tanítóanyag esetén is – minden tekintetben jelentősen növeli. Az eszköz nyílt forráskódú, Javában íródott, így működése könnyen módosítható. A rendszer alapjait a Brants [10] és Halácsy et al. [11] által ismertetett algoritmus

képezi, melyet úgy alakítottunk át, hogy képes legyen a morfológiai elemző integrált és hatékony használatára. Nagy előnye még a taggernek, hogy tanuló algoritmusának tanítási ideje – más maximum entrópia vagy CRF-alapú eljárásokhoz képest – nagyon alacsony, másodpercekben mérhető.

1. táblázat. A egyes szófaji egyértelműsítő modulok pontossága.

	PP	PP+	ME	PE	HuLaPos
Pontosság	83,82%	86,88%	80,14%	79,34%	81,59%

Az alábbiakban (1. táblázat) összehasonlítjuk a PurePos integrált HuMor morfológiai elemzőt tartalmazó változata (PP+), az integrált elemzőt nem használó (PP) és három további szófaji címkéző, az OpenNLP maximum entrópia (ME) és perceptronalapú taggere [12] (PE) és [13]-ban leírt, Moses dekoderen alapuló, Laki László által fejlesztett eszköznek (HuLaPos) a fenti tesztkorpuszon mért címkepontosságát. Valamennyi eszközt a Szeged korpusz Humor tagekre konvertált változatán tanítottuk be. A eszközök közül csak a PurePos és a HuLaPos ad lemmát is tartalmazó teljes morfológiai elemzést.

Egybevetve a szabadon elérhető nyelvfüggetlen PoS taggerek eredményességét, a PurePos hatékony eszköznek tűnik doménadaptációs eljárások fejlesztésére, tesztelésére. A korábbi és jelen eredmények alapján is elmondható, hogy a magyarhoz hasonlóan komplex morfológiájú nyelvek esetében a morfológiai tudás kulcsszerepet játszik egy magas pontosságú egyértelműsítőben, így szükségesnek ítéltük a HuMor morfológiai elemző orvosi doménre való adaptációját.

3.2. A morfológia adaptálása

Az egyértelműsítő rendszer egyik alkotóeleme a HuMor morfológiai elemző. Hogy az elemző orvosi szövegek elemzésében nyújtott teljesítményét növeljük, úgy határoztunk, hogy első körben lehetőleg viszonylag megbízható minőségű forrásból származó anyaggal bővítjük az elemző tőtárát.

A tőtár bővítésének egyik fontos forrása az 1992-ben megjelent Orvosi helyesírási szótár [14] volt. A helyesírási szótár semmiféle információt nem tartalmaz sem a benne szereplő szavak szófajára, sem azok nyelvére, illetve kiejtésére vonatkozólag, ezen információkra azonban a morfológiai adatbázisba való felvételükhöz szükség volt (illetve az összetett szavak esetében az összetételi határ helyét kellett megállapítanunk). Mivel több tízezer szót kellett annotálnunk, úgy döntöttünk, hogy a szavak kategorizálását és a hozzáadandó információk előállítását megpróbáljuk automatikus módszerekkel segíteni.

A szófaji kategorizációban egyrészt egyszerű formai jegyekre támaszkodhattunk (pl. a szótárban szereplő neveket és rövidítéseket ilyen alapon könnyen meg lehetett különböztetni az egyéb szavaktól). Másrészt a

szavak egy részének kézzel való szófaji kategorizációja után ezen az anyagon a PurePos-ban is alkalmazott végződésguesser-algoritmust tanítottuk be és alkalmaztuk, majd a kapott címkéket átnéztük és javítottuk, illetve ezt az eljárást iteráltuk. A latin-görög szókincs elemeinél bizonyos végződéstípusok esetében különösen nehéz volt eldönteni, hogy egy-egy szó főnév vagy melléknév, esetleg mindkettőként használatos. A kérdéses esetekben egyenként kellett utánanéznünk a szó jelentésének, illetve használatának, ami nagyon időigényes volt.

Ezért az automatikus szófaji osztályozásnál még egy szempontot figyelembe vettünk: a szótárban szereplő több tagú latinos kifejezések esetében az utolsó elem gyakran melléknév (hacsak nem birtokos szerkezetről van szó), a első elem pedig leginkább főnév, a elemek sorrendje tehát szisztematikusan különbözik a magyar jelzős szerkezetekétől. A latin melléknevek elsősorban emiatt jelentenek külön problémát a magyar nyelvű orvosi szövegek címkézése szempontjából. A magyarul írt megfelelőjük (amely a latin szó hímnemű alanyesetű alakjával áll alaki kapcsolatban) egyértelműen melléknév, amely a magyarban szokásos módon melléknév–főnév sorrendben áll. A valódi többszavas latin kifejezésekben a sorrend főnév–melléknév, és a két elem egyeztetve van. A nem hímnemű vagy esetleg nem alanyesetű szerkezetben álló latin melléknévi alakok a magyar címkézés szempontjából gyakorlatilag főnévnek tekinthetők. Elvileg ugyanez lenne a helyzet a hímnemű alanyesetűek szempontjából is, ha nem lenne a korpusz tele olyan szerkezetekkel, amelyek sorrendjükben a magyar névszói szerkezet mintáját követik (mivel azok), helyesírásukban azonban latinos írásmódú elemekből vannak összeállítva.

Ezért úgy döntöttünk, hogy a latin helyesírású főneveket és mellékneveket megkülönböztető címkével látjuk el a morfológiában, és ezek közül a hímnemű alanyesetű mellékneveket alapvetően melléknévként, a többit pedig főnévként címkézzük, hogy ha lesz elegendő kézzel ellenőrzött annotációt tartalmazó orvosi szöveget tartalmazó tanító anyagunk, a tagger ebből megtanulhassa a hímnemű alanyesetű latin melléknevek jellegzetes eloszlását. Sajnos a rendelkezésünkre álló idő egyelőre csak a tesztkorpusz létrehozására volt elegendő, ezért ezt a lehetőséget munkánk jelen fázisában nem tudtuk kihasználni, a latin szavakat megkülönböztető címkék tanító anyag híján egyelőre inkább problémát okoztak a taggernek, semmint segítséget.

A szófaj eldöntésén kívül tehát meg kellett különböztetnünk az idegen és a magyar helyesírású elemeket. Erre azért is szükség volt, mert az előbbiekhez a kiejtést is meg kellett határoznunk, hogy a szavak helyesen toldalékolódjanak. Ebben részben segítséget nyújtott, hogy a szótár utalásként sok olyan szópárt tartalmaz, amelyek ugyanannak a szónak vagy kifejezésnek a helyesírási változatai. Ezek legnagyobb részénél az egyik változat a magyar helyesírású, a másik az idegen helyesírású változat. Az esetek nagy részében a magyar volt preferált változatként megjelölve. Volt azonban az anyagban rengeteg kivétel is. Részleges manuális kategorizáció után erre a feladatra a TextCat algoritmus [15] egy adaptált implementációját használtuk, amely rövid stringekre is képes elég jól használható választ adni a magyar vagy nem magyar kérdésre. Viszonylag

egyértelmű volt a helyzet, ha egy szópár egyik tagját a rendszer inkább idegennek, a másikat pedig inkább magyarnak minősítette. A párok nagy része a szótárban ugyanakkor olyan, hogy mindkét eleme idegen, amelyek ugyanannak a szónak különböző írásváltozatai. Ezek kiszűrésében ugyancsak jó szolgálatot tett a fenti algoritmus. A korábban említett iteratív szótárbővítő eljárásnak ezt a nyelvmegállapító eljárást is részévé tettük. A szótár rengeteg olyan idegen (főleg görög-latin, emellett angol és francia) szót is tartalmaz, amelynek a magyar ortográfiával írt megfelelője nem szerepel a szótárban. Ezeket is fel kellett ismernünk, és itt nem támaszkodhattunk olyan implicit extra információra, amit a szópárok esetében a másik elem adott.

Amellett, hogy el kellett döntenünk, hogy az elem idegen vagy magyar, a konkrét kiejtést is hozzá kellett rendelni. Ez a hivatkozási rendszer folytán párban álló elemek esetében részben adott volt, bár az elemek nagy részének a magyaros mellett a latinos kiejtésére is szükségünk volt (különös tekintettel az s betűre végződő szavakra), hiszen sokszor önállóan is, több szavas latin frázis elemeiként viszont elvileg mindig a latinos kiejtés a mérvadó a toldalékolás szempontjából. Mivel rengeteg szó kiejtését kellett megadnunk, ezt sem kézzel csináltuk, hanem algoritmikusan állítottuk elő őket (az s végűeknél mindkét változatot), és az így előállított kiejtést javítottuk kézzel, ha szükséges volt. Erre a feladatra nem valamilyen általános gépi tanulás alapú G2P (grapheme-to-phoneme) algoritmust használtunk, hanem egyszerűen írtunk egy reguláris kifejezéseken alapuló heurisztikus algoritmust, amelynek kimenetét némi csiszolgatás után viszonylag keveset kellett javítgatni. Ezt akár a lexikon szerkesztésére használt editorból közvetlenül is meg lehetett hívni akár egy egyszerre kijelölt több szóból álló blokkra is, ha olyan szót találtunk, amelyet a korábbi algoritmusaink esetleg tévesen nem ítéltek idegennek.

További feladat volt az összetételi határok megállapítása, és az összetételekben gyakran szereplő elemek kiemelt kezelése: ezeket előrevettük a szavak feldolgozása során, így az ezeket tartalmazó összetételek kezelését a morfológiára bízva hatékonyabban csökkenthettük a feldolgozásra váró szótári tételek számát, illetve minimalizálhattuk a esetleges inkonzisztens manuális adatbevitel esélyét. Ehhez egy olyan algoritmust implementáltunk, amely az általános helyesírási szótárban és az orvosi helyesírási szótárban szóként szereplő legalább két karakter hosszú és magánhangzót is tartalmazó elemeket szófában eltárolva és azokat utótagként keresve a szótár szavaiban statisztikát készített az így felbontott szavak elemeiből, és a megtalált prefixumokat több szempontból osztályozta: külön megjelölte egyrészt a 4 karakternél rövidebbeket, a szótárban szóként létezőket, a belül kötőjelet tartalmazókat és azokat az eseteket, ahol a felbontott szó maga is utótagja volt a szótár valamelyik másik szavának. Ennek az eredményét felhasználva és a gyanúsnak tűnő elemekkel alkotott összetételeket külön kézzel ellenőrizve a leggyakoribb valódi elő- és utótagokat felvettük a szótárba, majd második körben az ezekkel képzett valódi összetételeket is, így hozzájutottunk a szótárban szereplő összetételek összetételi tagokat is jelölő reprezentációjához, amelyeket a szótárba felvettünk.

A szótár meglepő módon sok olyan igéből képzett szót (leginkább melléknévi igenevet és nomen actionist) tartalmaz, amelyek (általában latin-görög tőből képzett) alapigéje ugyanakkor nem szerepel benne. Ezek helyett a szavak helyett az alapigét vettük fel, hiszen így kapunk a képzett elemekre normális elemzést. A munka egyik fázisa az volt, amikor ezekre vadásztunk. Emellett sok olyan s-képzős melléknév szerepel a szótárban, amelyeknek alapszava is benne van. Első körben az ilyennek látszó szavakat is kihagytuk a feldolgozásból, mert az alapszó felvétele automatikusan a képzett szó bekerülését is jelentette. Ami még különös körültekintést indokolt a szótár feldolgozásakor, az az volt, hogy meglepően sok nyilvánvaló nyomdahibával találkoztunk benne, ezért nem lehetett készpénznek venni a szótárban szereplő adatokat.

A helyesírási szótár mellett a másik fontos feldolgozott szóanyag az OGYI[1] honlapjáról letöltött gyógyszernév- és hatóanyag-adatbázis volt. Itt a szavak kategorizálása és a szófaj eldöntése kevésbé okozott problémát. A kiejtés viszont itt is fontos volt. Az ezt kiszámoló algoritmusunkat annyiban adaptálnunk kellett, hogy mivel a hatóanyagok elnevezésére az jellemző, hogy bár azok alapvetően latinos-görögös elemekből épülnek fel, de az írásmódjuk az angolban szokásos képet mutatja, így a latin/görög végződések helyett, szinte mind ki nem ejtett *-e*-re végződik.

A szótárbővítés harmadik forrása természetesen maga a korpusz volt. Már a szótár feldolgozásakor előnyben részesítettük azokat a szavakat, amelyek a korpuszban is szerepeltek. De emellett az előbbi forrásaink feldolgozása után továbbra is elemzetlenül maradt gyakori szavak feldolgozása is fontos volt. Ezek túlnyomó része rövidítés volt. A gyakori rövidítések feloldását, és ez alapján a rövidítés szófaji besorolását (ha az nem volt a szóalak alapján teljesen nyilvánvaló) korpuszkonkordanciák alapján végeztük. Amire nem ügyeltünk eléggé (és ez nagyon jelentős negatív hatással volt a tesztek során a rendszer címkepontosságára), az az volt, hogy a feldolgozás során figyelmen kívül hagytuk azokat a pontra végződő szavakat (potenciális rövidítéseket), amelyre az elemzőnek már volt valamilyen elemzése, és így a korpuszban gyakori címkéjükkel a morfológiába nem kerültek bele.

Az orvosi szótár (egyelőre korántsem teljes) feldolgozása és a korpuszban szereplő leggyakoribb rövidítések felvétele együttesen 36000 tétellel bővítette a morfológia tőtárát (még mintegy 25000 szót nem dolgoztunk fel). A gyógyszernév-adatbázisból 4860 tétel került bele.

Az így javított elemzővel ellátott rendszer szófaji egyértelműsítésre számolt pontossága 93,25%, mellyel mintegy 6,4%-kal sikerült redukálni a korábbi rendszer hibáinak számát.

Közelebbről szemügyre véve a hibákat, azt tapasztaltuk, hogy a rendszer gyakori hibáinak egy része olyan jelenség, melyek a további szintaktikai, szemantikai feldolgozás szempontjából érdektelen. Ezek azon esetek, amikor a morfológia különbséget tesz latin, illetve magyar eredetű főnevek és melléknevek között, továbbá az igenevek és az ezekből lexikalizálódott melléknevek között.

[1] http://www.ogyi.hu/listak/

Ezen hibákat a továbbiakban nem számolva, a fenti eredmények 90,55%-ra és 93,77%-ra módosulnak az eredeti és a bővített morfológiát tekintve.

3.3. Az egyértelműsítő adaptálása

Az egyértelműsítő rendszer adaptálása során megoldandó első probléma az új, eddig a tanító anyagban nem látott címkék elérhetővé tétele a tagger lexikális és kontextuális modellje számára. A PurePos és minden más szófaji egyértelműsítő rendszer a tanulási fázisában a tanító anyagból a szófaji címke és a szó kontextusa alapján modellezi az adott szófaji kategória eloszlását. Így természetes módon, a tanítás során nem látott tagről semmilyen előzetes információval nem fog rendelkezni a modell. Mint ahogy azt a morfológia építésénél láttuk, a főnevek és melléknevek egy új kategóriáját vezettük be azon szavakra, melyek a latin morfológia szabályai szerint ragozandók. A morfológiához hozzáadott szavak jelentős hányadának csupán egyetlen elemzése van, s ha ez a fenti osztályok egyikébe tartozik, akkor bár az adott szóhoz ezen kategória fog tartozni, de az utána következő szavak címkézése során a kontextuális modell nem képes eloszlást rendelni. Továbbá, amikor egy szóhoz a HuMor több elemzést is ad, s ezek egyike egy újonnan létrehozott címke, akkor ehhez sem tartozik a megtanult modellek egyikében sem valószínűségi információ. Úgy találtuk, hogy a legjobb becslés, amit – egy új tanító anyag létrehozása nélkül – tehetünk, hogy a latin főneveket és mellékneveket a magyar főnevek eloszlásával becsüljük. (Így pl.: a *diagnosis* szó `[FN|lat][NOM]` címkéjét és a *sin.* szó `[MN|lat][NOM]` elemzését is az `[FN][NOM]` eloszlásával becsüljük.)

Az orvosi nyelvezet egyik sajátossága a rövidített szavak nagy mennyisége és változatos használata, nem beszélve ezek a normától különböző használatáról, helyesírásáról. Összehasonlításképpen: míg a Szeged Korpuszban a rövidítések a tokenek 0,36%-át teszik ki, addig az általunk javított anyag 8,49%-a rövidítés. Fontos különbség még, hogy ebben a speciális nyelvezetben az orvosok – eltérve a helyesírási normáktól – sokszor a toldalékokat nem kötik kötőjellel a rövidített szótőhöz, hanem egyszerűen lehagyják azt. (A tanító anyagban szereplő rövidítésekből a kötőjellel írottak aránya 9,36%, míg az egyértelműsítendőben 3,87%.) Pl.: a *jo*, *jo.*, *j. o.* „rövidítések" mindegyike a különböző kategóriájú *jobb oldal*, *jobb oldali*, *jobb oldalon* kifejezések bármelyikét jelentheti, az adott szövegkörnyezetben persze általában egyértelműen azonosíthatóan az egyikre utal.

A PurePos eredetileg sem a tanítási, sem pedig a címkézési fázisban nem kezeli különlegesen a rövidítéseket, mert egyrészt a norma szerint írott köznyelvi szövegeken a toldalékos alakok elemzésében nagy mértékben tud támaszkodni a POS-tageket tippelő suffix guesserre, másrészt általában nem kell ilyen mennyiségben és ennyire ad hoc módon létrehozott rövidítéstömeggel megbirkóznia. Ezzel a megközelítéssel jelen anyag esetén sokszor hibás következtetésre jut a tagger, így az alábbiak szerint módosítottuk a működését. A rendszer képes bizonyos előre definiált formai jegyeknek megfelelő szóalakokhoz külön lexikális eloszlást megtanulni, amit az alaprendszer a számjegyeket tartalmazó tokenekre, HTML entitásokra és írásjelekre alkalmaz. A fenti felsoroláshoz hozzáadtuk még

a toldalékolatlan alakú rövidítéseket, továbbá ezeket a tanítási fázisban elhagytuk a standard tokenekhez megtanult lexikális modellből. Így sikerült azt elérni, hogy a megtanult lexikális eloszlás ne az egyes tokenek eredetijéből fakadjon, hanem egy általánosabb, rövidítésekhez tartozóból. Mivel az adaptált morfológia számos rövidített alakot már ismer, ezért ezt a tudást is kívánatos volt alkalmazni. Az eredeti PurePos-ban a tanítóanyagban már látott szavak esetén az egyértelműsítő nem egyezteti a tanult tudást az integrált morfológiával, a rövidítések ilyen típusú kezelése, viszont szükségessé tette ezt. Az egyeztetés úgy történik, hogy a morfológia által javasolt latin típusú címkék a magyar megfelelővel való becsült valószínűséggel kerülnek be az egyértelműsítési folyamatba.

A fent bemutatott – a szófaji osztályok és a bizonyos tokenek reprezentációjának módosításával járó – doménadaptációs eljárással további javulást értünk el a taggelés területén, így 94,49%-os tokenszintű pontosságról számolhatunk be.

3.4. Hibaanalízis

Az összehasonlítási alapnak tekintett alaprendszer hibáit megvizsgálva, a hibákat az alábbi csoportokba lehet sorolni:

1. Az egyik leggyakoribb hiba, hogy a rövidítések hibás osztályba kerülnek, azok különleges írásmódja és nagyon változatos használata miatt. Ezen belül is tipikusan a főnévi és melléknévi szerepek keverése jellemző.
2. A hibák egy másik osztálya a latin, illetve latin eredetű kifejezések szófajának fentihez hasonló rossz meghatározása. Mivel ezen szóalakokat a korábban használt morfológiai elemző nem tudta megelemezni, így a guesserre maradt a feladat. A guesser rossz működése – a benne implementált tanulási algoritmus jellemzői miatt – nagyobbrészt a más doménen történő tanításból fakadnak.
3. A korpuszt alkotó orvosi szövegekben jellemző a melléknévi igenevek állítmányként történő használata, amely a köznyelvben meglehetősen ritka. Többek között ehhez kapcsolódóan a rendszer egyik gyakori hibaosztályát azok az esetek alkotják, amikor melléknévi igeneveket múlt idejű igékként annotál a rendszer. Ilyen tipikus rosszul elemzett szavak a *javasolt*, *kifejezett*, *igazolt*. Rendszeresen hibás analízist adott a PurePos a melléknévi igenév–melléknév ambiguitási osztály esetén is (pl.: *ismert*, *jelzett*). Hozzá kell tennünk, hogy ezeknek az eseteknek a megítélése a humán annotátorok számára is gyakran kétséges.
4. A fentieken kívül nagy számban vannak még jelen olyan hibák, melyek egyszerűen az orvosi nyelvhasználat egyediségéből fakadnak. Ilyen hibáson osztályozott szavak pl.: a *jobb*, mely a tanítóanyagban alapfokú melléknévként gyakorlatilag nem szerepel, vagy a *beteg*, melyet a tanulás során a PurePos soha nem látott főnévként. Ezen hibaesetek közös vonása, hogy a két korpuszban a kapcsolódó ambiguitási osztályok elemeinek eloszlása teljesen más.

Míg a 3.2 és 3.3 részekben részletezett megoldásokkal elsősorban az 1. és 2. típusú hibák javítását céloztuk meg, addig a 3. és 4. típusúak javításához szükségesnek látjuk a megtanult lexikai valószínűségek változtathatóságának a lehetőségét. Ehhez a továbbiakban úgy módosítjuk a PurePos rendszert, hogy a bemeneti mondatok egyes tokenjeinek elemzéseihez a címkézési folyamat segítésére a felhasználó által előredefiniált eloszlást rendelhessünk. Így a rendszer képessé válhat arra, hogy néhány egyszerű szabályt használva, nagyon gyakori tévesztések célzott javításával kis erőfeszítéssel nagy mértékben javítsuk az annotálás pontosságát. További tervünk, hogy a korpusz mellett további egyéb orvosi adatbázisokat is felhasználva olyan rövidítésfeloldó rendszert hozzunk létre, amely különösen a több elemből álló rövidítések esetében a jelenleginél jóval nagyobb pontossággal képes a rövidített szavak címkézésére.

4. Összegzés

Írásunkban ismertettük egy folyamatban lévő kutatási projekt aktuális állását, melynek részeként bemutattuk a rendelkezésünkre álló orvosi rekordokon végzett azon előfeldolgozási lépéseket, amelyeket szükségesnek véltünk egy gold standard korpusz létrehozásához. Azt is láttuk, hogy az így létrehozott eszközök egy későbbi orvosi rekordokra épülő szövegbányászati rendszer fontos építőkövei lehetnek. Bemutattuk azon lépéseket, amelyekkel a HuMor morfológiai elemzőt az orvosi doménre adaptáltuk, továbbá megvizsgáltuk, hogy az így előállt megnövekedett morfológiai tudást mily módon lehetséges mélyebben integrálni a PurePos morfológiai egyértelműsítő rendszerbe. Részletes hibaanalízist végeztünk, s a felmerülő hibák egy részére teljes, illetve részleges megoldást mutattunk be.

A jövőben folytatjuk a rendszer doménadaptálását, s ennek keretében a rövidítések kezelésére bevezetünk egy olyan alrendszert, mely prefixegyezés alapján statisztikai módszerrel próbálkozik a rövidítések feloldásával, hogy az azokhoz tartozó lexikális eloszlást a rövidített szó eredetijéből nyerjük ki. Célunk még, hogy folytassuk a manuális annotálást, hogy a PurePos elemzővel végzendő további doménadaptációs kísérletekhez megfelelő tanítóanyag is rendelkezésünkre álljon, illetve hogy korábban semmilyen szempontból sem látott tesztanyagon is validálhassunk eredményeinket.

Hivatkozások

1. Siklósi, B., Orosz, Gy., Novák, A., Prószéky, G.: Automatic structuring and correction suggestion system for Hungarian clinical records. In De Pauw, G., de Schryver, G.M., Forcada, M.L., M. Tyers, F., Waiganjo Wagacha, P., eds.: 8th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages, Istanbul (2012) 29–34
2. Siklósi, B., Orosz, Gy., Novák, A.: Magyar nyelvű klinikai dokumentumok előfeldolgozása. In Tanács, A., Vincze, V., eds.: Magyar Számítógépes Nyelvészeti Konferencia 2011, Szeged (2011) 143
3. Novák, A.: Milyen a jó humor? In: Magyar Számítógépes Nyelvészeti Konferencia 2003, Szeged (2003) 138–145

4. Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Inquiries into Words, Constraints and Contexts., Stanford, California (2005) 150–157
5. Quinlan, J.R.: C4.5: Programs for Machine Learning. Volume 1 of Morgan Kaufmann series in Machine Learning. Morgan Kaufmann (1993)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. ACM SIGKDD Explorations Newsletter **11**(1) (2009) 10
7. Miháczi, A., Németh, L., Rácz, M.: Magyar szövegek természetes nyelvi előfeldolgozása. In: Magyar Számítógépes Nyelvészeti Konferencia 2003, Szeged (2003) 38–43
8. Orosz, Gy., Novák, A.: PurePos – an open source morphological disambiguator. In Sharp, B., Zock, M., eds.: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science, Wroclaw (2012) 53–63
9. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004. (2004) 19–23
10. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the sixth conference on Applied natural language processing. Number i, Universität des Saarlandes, Computational Linguistics, Association for Computational Linguistics (2000) 224–231
11. Halácsy, P., Kornai, A., Oravecz, C.: HunPos: an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Prague, Czech Republic, Association for Computational Linguistics (2007) 209–212
12. Baldridge, J., Morton, T., Bierner, G.: The OpenNLP maximum entropy package (2002)
13. Laki, L.J.: Investigating the Possibilities of Using SMT for Text Annotation. In: SLATE 2012 - Symposium on Languages, Applications and Technologies, Braga, Portugal, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012) 267–283.
14. Fábián, P., Magasi, P.: Orvosi helyesírási szótár. Akadémiai Kiadó, Budapest (1992)
15. Cavnar, W.B., Trenkle, J.M.: N-Gram-Based Text Categorization. Ann Arbor MI **48113**(2) (1994) 161–175

O & kǫzèpmaǵar zoalactanÿ èlèmzǫ

Novák Attila[1,2], Wenszky Nóra[2]

[1]MTA Nyelvtudományi Intézet
1068 Budapest, Benczúr utca 33.
[2] MTA–PPKE Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter utca 50/a
novak@nytud.hu

Kivonat: Cikkünkben egy olyan magyar számítógépes morfológiát mutatunk be, amelyet kiegészítettünk az ómagyarban és a középmagyarban még létező, de azóta kihalt alaktani szerkezetek leírásával, illetve a szükséges szókinccsel, így alkalmas régi magyar szövegek elemzésére. Az elemzőt két, a Nyelvtudományi Intézetben párhuzamosan futó, ómagyar, illetve középmagyar szövegek feldolgozásával foglalkozó OTKA kutatási projektben használjuk. A morfológia mellett bemutatjuk a szövegek morfoszintaktikai annotálására használt gépi és kézi egyértelműsítő rendszert, valamint az annotált szövegekben való keresést lehetővé tevő korpuszkezelőt.

1 Bevezetés

A Nyelvtudományi Intézet két OTKA projektjének (Magyar generatív történeti szintaxis [OTKA NK78074], valamint Történeti magánéleti korpusz [OTKA 81189]) feladata többek között az ómagyar és a középmagyar időszakból származó szövegeket tartalmazó morfológiailag elemzett, kereshető korpuszok létrehozása. A projektekben a Humor magyar morfológiai elemző [7] olyan kibővített változatát használjuk, amelyet alkalmassá tettünk a nyelvből időközben kihalt alaktani konstrukciókat, toldalékallomorfokat, toldalékmorfémákat, paradigmákat, töveket tartalmazó szavak elemzésére is. Az alábbiakban áttekintjük az elemzőprogram kifejlesztéséhez szükséges lépéseket, a felmerülő problémákat és megoldásukat, valamint a szövegek morfoszintaktikai annotálására használt gépi és kézi egyértelműsítő rendszert és az annotált szövegekben való keresést lehetővé tevő korpuszkezelőt.

2 A szövegek előfeldolgozása

Mindkét szóban forgó projektnek – a középmagyar szövegekkel foglalkozónak kizárólagos – célja, hogy annotált, kereshető korpuszokat hozzon létre. Míg az ómagyar korból főként kódexek maradtak fenn, és a szövegek nagy része fordítás, a középmagyar korpusz elkészítésekor a célkitűzés az élő nyelvhez sokkal közelebb álló források összeválogatása volt. Így ezt a korpuszt perszövegek – közöttük boszor-

kányperek jegyzőkönyvei – és misszilisek, azaz ténylegesen elküldött főúri és jobbágylevelek alkotják. Az utóbbi korpusz esetében az egyes szövegekhez tartozó metaadatok is fontos szerepet játszanak, amelyek lehetővé teszik ezeknek a forrásoknak történeti-szociolingvisztikai szempontú vizsgálatát is.

2.1 Digitalizálás

A korpuszokat alkotó szövegek eredetileg kéziratos formában maradtak fenn, azonban egyik projektnek sem képezte részét kéziratos szövegek feldolgozása: minden esetben nyomtatott szövegkiadásokból dolgoztunk. A szövegek nagy részének az esetében azonban nem állt rendelkezésre digitalizált szövegváltozat. Így az első feladat a szövegek digitalizálása volt, amelyet az esetek többségében OCR alkalmazásával végeztünk el. Különösen az ómagyar időszakból származó szövegek esetében jelentett nehéz feladatot a szokatlan karakterek és mellékjel-kombinációk feldolgozása. Minden egyes szöveghez újra be kellett tanítani az alkalmazott OCR programot, hiszen másmás különleges karakterek szerepeltek bennük. Az automatikusan felismertetett szövegben azonban így is számos hiba maradt, munkatársainknak tehát minden szöveget végig kellett olvasni. Az eredeti, kinyomtatott szöveget és a digitalizált változatot össze kellett hasonlítani és a beviteli hibákat kézzel javítani.

EVANGELIVM SECVNDVM MATTHAEVM
|| [I]
8ra Jeſus criſtus dauid fia abracham fia zuletetenec ꝙnuꝙ (2) Abracham ke · zule iſaakot' Iſaac ke · zule iacobot Jacob ke · zule iudaſt2 ꝙ attafiait (3) Judas ke · zule phaꝛeſt' 2 Zaramot thamaꝛtol' Phares ke · zule Ezromot Ezꝛom ke · zule Aramot (4) Aram ke · zule Aminadabot Aminadab ke · zule Naaſſont

2.2 Normalizálás

A szövegek rendkívül változatos írásképe, az előforduló sokféle dialektus, illetve az átfogott hosszú időszak folyamán bekövetkezett nagymérvű nyelvtörténeti (elsősorban fonológiai) változások miatt az automatikus elemzés egyik feltétele a szövegek írásképi és fonológiai szempontból egységes formára hozása, azaz normalizálása volt. Ez nagyrészt kézzel történt, és a folyamat során a szövegeket tagmondatokra is bontottuk. A projektben nem volt célunk, hogy olyan elemzőt hozzunk létre, amely a korpuszt alkotó eredeti szövegek teljes fonológiai dialektális változatosságát kezeli. Így a normalizálás során az ilyen jellegű különbségeket – például az *ö*-zést – eltüntettük.

höti	után	vallia
hite	után	vallja:
hit	után	vall
N.PxS3	PP	V.S3.Def

szőrö	szalát	sem	fogta	el,
szőre	szálát	sem	fogta	el.
szőr	szál	sem	fog	el
N.PxS3	N.PxS3.Acc	Adv	V.Past.S3.Def	VPfx

Fontos szempont volt azonban az, hogy morfémák a normalizálás folyamán ne tűnjenek el vagy alakuljanak át más morfémákká: például az elbeszélő múltban álló alakokat nem alakítottuk egyszerű múlt időkké stb. A morfémahűség helyes megvalósításához általában alaposan mérlegelnünk kellett az adott korszak ortográfiájának jellegzetességeit. Törekedtünk rá, hogy a korabeli helyesírás bizonytalanságaiból adódó inherens és ténylegesen feloldhatatlan többértelműségeket lehetőleg ne tüntessük el a normalizálás során.

Az egyik jellegzetes többértelműség a korai szövegek magánhangzóhosszúság-jelölésének hiányából, illetve bizonytalanságából és abból a tényből adódott, hogy a határozott tárgyas igeragozás használatának szabályszerűségei az adott időszakban részben különböztek attól, amit a szöveget normalizáló nyelvészek anyanyelvi intuíciója esetleg sugallna. A szövegek egy részében például egyértelműen megfigyelhető, hogy egyenes idézés esetén – ellentétben a mai köznyelvben szokásostól – a *mond* ige határozatlan ragozással is használatos volt.

mondotta	a	Felesegenek
mondta	a	feleségének:
mond	a	feleség
V.Past.S3.Def	Det	N.PxS3.Dat

arra	**mond**	Lovasz	Matyasne
Arra	**mond**	Lovász	Mátyásné:
az	**mond**	Lovász	Mátyásné
N|Pro.Sub	**V.S3**	N	N

Az elbeszélő múltban azonban a *monda* igealak ebben a helyzetben magánhangzóhosszúság-jelölésének bizonytalansága miatt éppoly kevéssé rekonstruálható módon utal az igeragozás határozott vagy határozatlan voltára *(monda ~ mondá)*, mint a *mondtam* alak. A bizonytalanság forrása itt a rag magánhangzója hosszúságának bizonytalanságából fakad, amelyet a normalizált szövegben ilyen esetben a magánhangzó után írt ékezettel jelölünk.

monda	erre	Giöröfy	Janosne,
Monda´	erre	Györöfy	Jánosné:
mond	ez	Györöfy	Jánosné
V.Ipf.S3.Def?	N|Pro.Sub	N	N

én	**mondottam**	néki	
Én	**mondtam**	neki:	
én	**mond**	ő	
N|Pro.S1	**V.Past.S1.Def?**	N|Pro.Dat.S3	

Hasonlóan bizonytalan az igeragozás határozott volta abban az esetben, ha a tárgy birtokos szerkezet, de nincs definit determinánsa. Ebben az esetben a határozott vagy határozatlan igeragozás használata dialektusfüggő. (Az alábbi példákban a *nyavalyáját* determinánsa a szintén dialektusfüggő definitségű *mely*, a többi birtokos tárgy pedig determináns nélküli). A szöveget normalizáló vagy annotáló személy ilyenkor nem vetítheti a saját intuícióját az adott szövegre. Alább az első két példa a szerzők számára agrammatikus, mert a birtokos szerkezet tárgy mellett mindenképp definit igeragozást használnánk. Azonban mivel tudjuk, hogy más dialektusokban ez nem feltétlenül van így, az elbeszélő múltat tartalmazó harmadik szerkezetet inherensen többértelműnek kell tartanunk, nem tudván, hogy melyik dialektusból származik.

mely	**nyavalájáát**	Tormánéra	**gyanétott,**
mely	**nyavalyáját**	Tormánéra	**gyanított,**
a+mely	**nyavalya**	Tormáné	**gyanít**
Det|Pro|Rel	**N.PxS3.Acc**	N.Sub	**V.Past.S3**

hogy	holt	Ember	**koponyáit**	az	Padlason	**tartott**	volna,
hogy	holt	ember	**koponyáit**	a	padláson	**tartott**	volna,
hogy	holt	ember	**koponya**	a	padlás	**tart**	van
C	Adj	N	**N.PxS3.Pl.Acc**	Det	N.Sup	**V.Past.S3**	V.Cond

azulta	**halla**	rosz	**hirét,**	és	nevét,
azolta	**halla´**	rossz	**hírét**	és	nevét,
azolta_az+óta	**hall**	rossz	**hír**	és	név
Adv|Pro	**V.Ipf.S3.Def?**	Adj	**N.PxS3.Acc**	C	N.PxS3.Acc

Hasonló rendszeres többértelműségek jelentkeznek az elöl képzett tövek i-ző birtokos alakjai esetében, ha egyéb rag is van a szó végén (pl. *cselekedetinek*). Ezekben az esetekben még a szövegkörnyezet alapján sem mindig lehet egyértelműen eldönteni, hogy egyes számú vagy többes számú alakról van szó (*cselekedetének* vs. *cselekedeteinek*). Ilyenkor a normalizálás során meghagyjuk az i-ző birtokos alakot, az elemzőt pedig képessé tettük arra, hogy ezeket a szóalakokat úgy is tudja elemezni hogy a számot bizonytalannak jelöli:

csupán	az	Aszony	**cselekedetinek**	tulajdonitotta,
csupán	az	asszony	**cselekedetinek**	tulajdonította.
csupán	az	asszony	**cselekedet**	tulajdonít
Adv	Det	N	**N.PxS3.Pl?=i.Dat**	V.Past.S3.Def

Egyes szövegek korábbi normalizálása nem az általunk lefektetett elvek szerint történt, ilyen volt pl. a Székelyudvarhelyi kódex. Ennek szövege a mai magyar helyesírásnak megfelelő hangjelölést alkalmaz, azonban a szöveg fonológiai-dialektális sajátosságait nem közelítették a mai magyarhoz, ezért további kézi adaptációra volt szükség.

2.3 A -bAn/bA probléma

A normalizálás és a különösen a morfémahűség megítélése szempontjából speciális problémát jelentett a -bAn, illetve -bA ragos szóalakok kezelése. A két korpusz szövegeinek vizsgálata egyértelműen azt jelzi, hogy a két ragnak a beszélt nyelvben jelenleg sem éles szétválása sok száz éve stabilan fennálló állapot [6] (nevezetesen, hogy a -bA változat szóban minden további nélkül használható a -bAn funkciójában is, miközben az utóbbi változat is létezik és használatos), amely a leírt szövegekben általában meglehetősen zavaros képhez vezetett. A korpusz szövegei egyértelműen jelentősen különböznek abból a szempontból, hogy a feltételezhetően inesszívusz, illetve illatívusz funkciójú elemek jelölésére mennyire következetesen melyik ragalakot írták le. A -bAn/-bA elemeket tartalmazó szóalakok ortográfiája szempontjából merőben különböző megoldásokat találunk a korpuszban, még két egymással apa–fia relációban álló személy (Nádasdy Tamás és Nádasdy Ferenc) esetében is (az előbbi szinte kizárólag a -bA alakot, az utóbbi szinte kizárólag a –bAn-t használja minden funkcióban).

Azért, hogy biztosan ne essünk se abba a hibába, hogy egy merőben ortográfiai ügyet grammatikainak hiszünk, és így hibás elemzések tömkelegét állítjuk elő, se abba, hogy visszakövethetetlen módon mindent átírunk a saját kompetenciánknak

megfelelő alakra, azt a megoldást választottuk, hogy a -bAn/-bA elemeket tartalmazó szóalakok normalizálása során explicite jelöltük azokat az eseteket, ahol mindent a lehető leggondosabban mérlegelve úgy ítéltük, hogy a leírt alak nem felel meg a szándékolt grammatikai funkciónak, illetve az általunk használt ortográfiai normának, így a normalizált alak és az elemzés alapján visszakereshetők és kvantifikálhatók az egyes szövegek a -bAn/-bA-jellemzői.

az	Macska	az	Tehent	szopja	az	olba
A	macska	a	tehént	szopja	az	ólba'."
a	macska	a	tehén	szop	az	ól
Det	N	Det	N.Acc	V.S3.Def	Det	N.Ine

azomba	Bekene	aszt	mondotta
Azonba'	Bekéné	azt	mondta:
azonban	Bekéné	az	mond
C	N	N|Pro.Acc	V.Past.S3.Def

az	asztal	melől,	a	tüz	eleiben	ment,
Az	asztal	mellől	a	tűz	eleibe'n	ment,
az	asztal	mellől	a	tűz	eleibe_elébe	megy
Det	N	PP	Det	N	PP	V.Past.S3

azomban	midőn	bé ment	az	Orvos	Házaban
azonban	midőn	bement	az	orvos	házába'n,
azonban	a+midőn	be|+megy	az	orvos	ház
C	Adv|Pro|Rel	V.Past.S3	Det	N	N.PxS3.III

2.3 Jakab-féle adattárak

Az ómagyar kódexek egy része (a Jókai- [2], a Guary- [3], az Apor- [4] és a Festetics-kódex [5]) szótárszerű formában számítógépes nyelvtörténeti adattárként Jakab László debreceni kollektívája által feldolgozva volt elérhető. Ezekből az 1978 és 2002 között készült kiadásokból igen komoly erőfeszítést igényelt a szövegek visszaállítása. Bár ezek kézzel készült elemzést tartalmaztak, az nehezen olvasható numerikus kódok formájában szerepelt. Az olvashatatlan reprezentációból következő módon gyakran hibás, hiányos, ezen kívül – elsősorban a zárt szóosztályok elemei esetében – az általunk használt elemzésekkel inkompatibilis volt. Ennek ellenére sikerült a szövegeket a szótárakból visszaállítani, az elemzéseket konvertálni és kiegészíteni, ezek alapján automatikusan normalizált változatot generálni, és azt újraelemezni.

A Jakab-féle szótárszerű kiadásokban a szavak az eredeti kódexbeli előfordulásuk helyét (locusát) az oldal/kolumna és az azon belüli sorszám szintjén adták meg. Az alábbi részlet a Jókai-kódex szótárkiadásából származik.

080/08	**ablak**	ablakba	0002	000000	02	11	000	00	05	01
180/15	ablak	ablakbalol	0002	000000	02	11	000	00	09	01
109/12	ablak	ablakokba	0002	000000	02	11	000	01	05	01
159/03	ablak	ablakarol	0000	000000	02	11	000	13	17	01
126/08	**ábráz**	abraz	0000	000000	02	41	000	00	00	01
125/26	ábráz	abrazban	0000	000000	02	41	000	00	08	01
130/22	**abrosz**	Abroz	0000	000000	02	11	000	00	00	01
083/20	abrosz	abrozokott	0003	200000	02	11	000	01	01	01
034/24	**ad**	ad	0000	000000	01	11	000	00	06	01
062/15	ad	ad	0000	000000	01	11	000	00	06	01
082/19	ad	ad	0000	000000	01	11	000	00	06	01

A gyakori szavaknak nem minden előfordulása szerepel ténylegesen a szótári részben. Egy külön függelékben elemzés nélkül fel vannak sorolva az egyéb előfordulá-

sok és írásváltozatok, amelyek közül szerencsés esetben az egyiknél az elemzés is megtalálható. A függelék formája következményeként egyetlen hiba szóelőfordulások tucatjainak rossz elemzését eredményezhette, és eredményezte is.

UTÁN ~ UTÁNA

8/6, 38/8, 63/3, 101/13, 105/14, 106/1, 107/1, 122/7, 132/20, 143/27, 156/7, *vtan* 14/22, 24/25, 62/8, 99/16, 109/26, 120/1, 122/14, 160/26, *vttan* 143/8 *(20 adat)*
18/22, 22/24, 76/17, 90/2, 98/6, 101/8, 106/24, 130/7, 148/10, 160/26, *uttanna* 39/13, 79/14, 132/14, *uta[n]na* 38/22, 101/14, *vtanna* 7/25, 15/17, 25/23, 24, 51/17, 78/10, 138/14, 144/26, 150/16, *vttanna* 57/23 *(25 adat)*

(Összesen: 45 adat)

Az egyes sorok szavainak sorrendjét kézzel kellett a nyomtatott kiadás segítségével helyreállítani. A munkát némileg nehezítette, hogy ugyanabban a sorban néha többször szerepelt ugyanaz a szó – esetleg különböző elemzéssel, de ezekben az esetekben a szótárban általában csak egy előfordulás volt megadva.

003/15	mond	**Monda**	0	1	**11**	1	**13**	**0**	1	0
003/15	ön	ewn	0	6	11	200	0	4	1	0
003/16	jonh	yonhanban	0	2	21	0	13	8	1	3
005/17	s	s	0	10	11	0	0	0	0	0
005/17	mond	**monda**	0	1	**11**	1	**10**	**6**	1	0
005/17	atyjafia	**Attyamfÿa**	100	2	**12**	0	**13**	**0**	3	9
005/18	Ferenc	ferenc	0	3	11	0	0	0	1	0
006/10	de1	De	0	10	11	0	0	0	0	0
006/10	úr	vr	0	2	11	0	0	0	2	0
006/10	Bernald	bernald	0	3	21	0	0	0	1	0
006/10	mond	**monda**	0	1	**11**	1	**12**	**20**	1	3

A visszaállított szövegek számkódos morfológiai elemzéseit programmal konvertáltuk olvasható – és amennyire lehetséges volt – az időközben elkészült morfológiai elemző címkéivel kompatibilis elemzésekké. Ezekre az elemzésekre a morfológiát generátorként alkalmazva megkaptuk a szavak normalizált alakját is.

Ezeket az eredeti szóalakokkal összevetve alább világosan látszanak azok az esetek, ahol a szótárkiadásban hibás elemzés szerepelt, vagy esetleg a feldolgozás során került valamilyen hibás adat az anyagba. Alább az 5/17 *atyámfia* helyett az *atyjafia*, illetve a 6/10 *mondá* vagy *monda* (ez éppen a korábban említett kérdéses definitségű szóalak) helyett a *mondám* szóalak elemzése – ez a hiba a szóalak gyakorisága folytán a szótár függelékében megadott hivatkozás hibás feloldása miatt 106 szóalakot érintett a Jókai-kódexben. Szerencsére ez a hiba könnyen javítható volt.

003 15	Monda	**mondá**	**mond[V.Ipf.S3.Def]**
003 15	ewn	ön	ön[N|Pro.Nom_gen]
003 16	yonhanban	jonhában	jonh[N.PxS3.Ine]
005 17	s	s	s[C]
005 17	monda	**monda**	**mond[V.Ipf.S3]**
005 17	Attyamfÿa	**atyjafia**	**atyjafia[N.PxS3]**
005 18	ferenc	Ferenc.	Ferenc[N]
006 10	De	de	de[C]

006 10	vr	úr	úr[N]
006 10	bernald	Bernald	Bernald[N]
006 10	monda	**mondám**	**mond[V.Ipf.S1.Def]**

A kigenerált szóalakokat eztán újraelemeztük, mert az adattárban megadott elemzések egy része hiányos, illetve az elemző által visszaadott elemzésekkel inkompatibilis volt (elsősorban a névmások és az igenevek esetében). A kapott elemzések közül az adattárban megadotthoz leghasonlóbbat választottuk. Az alkalmazott hasonlósági mérték a trigramhasonlóság volt, amelyet meghatározott heurisztikus konverziók után alkalmaztunk.

A Jakab-féle kódrendszer legsúlyosabb hiányossága az volt, hogy az igenevek fajtáit és ragozott alakjait az általuk használt kódrendszer nem különböztette meg. Ezért ezeket a szavakat és a valódi elemzésüket a program az eredeti ómagyar írásmódú szóalakot is figyelembe véve különböző heurisztikákra alapozva próbálta rekonstruálni. Az alábbi tagmondatban például három szóalak (*p[ro]phetalo, vilagossolot, lattuan*) is igenévként szerepel (14-es kód), de semmi egyéb információ nem derül ki a kódokból sem az igenév fajtájára, sem az esetleges további ragokra vonatkozólag.

005/02	de	De	0	10	11	0	0	0	0	0
005/02	prófétál	**p[ro]phetalo**	0	**14**	11	120	**0**	**0**	10	100
005/02	lélek	lelekuel	4000	2	11	2	0	19	4	0
005/03	világosul	**vilagossolot**	100302	**14**	21	522	**0**	**0**	1	1
005/03	eleve	eleue	0	7	11	0	0	29	0	5
005/03	lát	**lattuan**	0	**14**	11	20	**0**	**0**	0	5
005/03	nagy	nagÿ	0	7	31	0	0	0	0	0
005/03	gond	gondokot	200000	2	11	0	1	1	1	0

A szövegen a fent leírt transzformációkat alkalmazva az alábbit kaptuk:

005	02	De	de	de[C]
005	02	p{ro}phetalo	**prófétáló**	**prófétál[V.PartPrs]**
005	02	lelekuel	lélekkel	lélek[N.Ins]
005	03	vilagossolot	**világosult**	**világosul[V.PartPrf]**
005	03	eleue==	eleve	eleve[Adv]
005	03	lattuan	**látván**	**lát[V.PartAdv=vÁn]**
005	03	nagÿ	nagy	nagy[Adv]
005	03	gondokot	gondokat	gond[N.Pl.Acc]

Az így automatikusan generált szöveget ezután még kézzel ellenőrizni kellett.

3 A morfológiai elemző

A digitalizált és normalizált szövegek elemzésére a Humor magyar morfológiai elemző [7] egy erre a célra kibővített változatát alkalmaztuk. Ehhez ki kellett bővíteni a program tőtárát és toldaléktárát az időközben kihalt paradigmákkal, szótövekkel és toldalékokkal, illetve toldalékallomorfokkal. Az alábbiakban az utóbbiakra láthatunk példákat (félkövérrel kiemelve).

sok	ỹdeyglen	ezkeppen	Asisia	altal	menuen
sok	**ideiglen**	ekképpen	Assisia	által	menvén
sok	**idő**	ekképpen	Assisia	által	megy
Q	**N.Ter=iglAn**	Adv|Pro	N	PP	V.PartAdv=vÁn

Es	yme	efelkoron	kezde	az	hews	asoytanỹa
és	íme	**éjfélkoron**	kezde	az	hős	ásítania,
és	íme	**éjfél**	kezd	az	hős	ásít
C	Int/Utt	**N.Tem=koron**	V.Ipf.S3	Det	N	V.Inf.S3

hoǵ ha	ki	ne{m}	monthatnaia	az	oldalann	is	ki	facadna
hogyha	ki	nem	**mondhatnája**	az	oldalán	is	ki	fakadna.
hogy+ha	ki	nem	**mond**	az	oldal	is	ki	fakad
C	VPfx	Adv	**V.Mod.Cond.S3=nÁjA.Def**	Det	N.PxS3.Sup	C	VPfx	V.Cond.S3

Es	zent	ferencznek	alazatost	engeduen
és	szent	Ferencnek	**alázatost**	engedvén
és	szent	Ferenc	**alázatos**	enged
C	Adj	N.Dat_gen	**Adj.Essmod=t**	V.PartAdv=vÁn

mert	ha	mindọn	iozagoncat	oztogatnoc	istenert	zegeńọknec
mert	ha	minden	jószágunkat	**osztogatnók**	istenért	szegényeknek
mert	ha	minden	jószág	**osztogat**	isten	szegény
C	C	N|Pro	N.PxP1.Acc	**V.Cond.P1=nÓk.Def**	N.Cau	N.Pl.Dat

A elemző toldaléktárába 50 új toldalékot vettünk fel (ezek alakváltozatait, allomorfjait nem számolva). Az alábbiakban olyan toldalékmorfémákra láthatunk példákat az igei morfológia köréből (félkövérrel kiemelve), amelyek a mai magyarban már nem használatosak.

Es	marada	a	nẽbẽṙi	ọ	kèt	fiaytol	&	ferietọl	meǵvalt·
és	**marada**	a	némber	ő	két	fiaitól	és	férjétől	megvált.
és	**marad**	a	némber	ő	két	fiú	és	férj	meg|+válik
C	**V.Ipf.S3**	Det	N	N|Pro.S3	Q	N.PxS3.Pl.Abl	C	N.PxS3.Abl	V.PartPrf

Azért	èlmènè	ember	ludanac	Bètlèhèmé~~bèn~~bọl
azért	**elmene**	ember	Júdeának	Betleheméből,
azért	**el|+megy**	ember	Júdea	betlehem
Adv|Pro	**V.Ipf.S3**	N	N.Dat	N.PxS3.Ela

lọn~~[èf]~~	èhſéǵ	a	fọldọ
lőn	éhség	a	földön.
lesz	éhség	a	föld
V.Ipf.S3	N	Det	N.Sup

bátyádat	oda	affelé	bocsátók	két ezer	emberrel.
Bátyádat	oda,	afelé	**bocsátók**	kétezer	emberrel,
báty	oda	a|+felé+	**bocsát**	két+ezer	ember
N.PxS2.Acc	Adv|Pro	Adv|Pro	**V.Ipf.P1.Def**	Q	N.Ins

Es	valobizoń	azt	nem	foghatoc	meg
és	valóbizony	azt	nem	**foghatók**	meg,
és	való+bizony	az	nem	**fog**	meg
C	Adv	N|Pro.Acc	Adv	**V.Mod.Ipf.P1.Def**	C

A me'l	fọld	tègedèt	m̃ghalauad	fogadād
amely	föld	tégedet	**meghalvád**	**fogadand,**
a+mely	föld	te	**meg|+hal**	**fogad**
N|Pro|Rel	N	N|Pro.S2.Acc	**V.PartAdv.S2**	**V.Fut.S3**

Bizonyos toldalékok produktivitása jelentősen csökkent az ómagyar korhoz képest. Bár ezekkel alkotott szavak a mai magyarban is léteznek, de csak lexikalizálódott egységekként, gyakran az eredetitől különböző szemantikával. Ilyen például az *-At*

képző, ami eredetileg a nomen actionis képző szerepét töltötte be, és teljesen produktív volt. Ennek szerepét vette át később az *-Ás* képző. Jelenleg a cselekvés tárgyi eredményét jelöli (nomen facti, pl. *épület, falazat*) – már ha a szó egyáltalán létezik.

Arra vonatkozólag, hogy az egyes toldalékoknak mely alakváltozatai a töveknek mely alakváltozataihoz kapcsolódtak, tehát hogyan alakultak a paradigmák, nemigen találtunk jól használható leírást. Az adatokat sokszor magukból a forrásokból kellett kideríteni. Bizonyos, időközben kihalt alaktani konstrukciókra viszonylag kevés adat van (pl. az alábbi egyeztetett határozói igenevekre), ráadásul a paradigmák számos elemére sokszor van egyéb lehetséges elemzés is. Ezek formális leírása ezért néha komoly kihívást jelentett.

èl	kẹd	mv́	vrõc	iſtènọ̃c	mèt	ọ	Angala	mègọrizọt	ènget	innèt
él	kedig	mi	Urunk	Istenünk,	mert	ő	angyala	megőrzött	engem	innét
él	kedig	mi	Ur	isten	mert	ő	angyal	meg|+őriz	én	innét
V.S3	C	N|Pro.P1	N.PxP1	N.PxP1	C	N.Nom_gen	N.PxS3	V.Past.S3	N|Pro.S1.Acc	Adv|Pro

èlmènèttèm	es ·	ot	**lakattam**	es ·	&	onnat	idè	fordolattam
elmenettem	is,	ott	**lakattam**	is,	és	onnan	ide	fordulattam.
el|+megy	is	ott	**lakik**	is	és	onnan	ide	fordul
VPfx.V.PartAdv=AttA.S1	Adv	Adv|Pro	**V.PartAdv=AttA.S1**	Adv	C	Adv|Pro	Adv|Pro	V.PartAdv=AttA.S1

lők	lől	lőn
lesz[V.Ipf.S1]	lesz[V.Ipf.S2]	lesz[V.Ipf.S3]
lőnk	lőtök	lőnek
lesz[V.Ipf.P1]	lesz[V.Ipf.P2]	lesz[V.Ipf.P3]

fekvém	fekvéd	fekvén
fekszik[V.PartAdv.S1]	fekszik[V.PartAdv.S2]	fekszik[V.PartAdv=vÁn]
fekvénk	fekvétek	fekvéjük
fekszik[V.PartAdv.P1]	fekszik[V.PartAdv.P2]	fekszik[V.PartAdv.P3]

A toldalékok és paradigmák leírásánál nagyságrendileg több munkát jelentett azoknak a töveknek a felvétele, amelyek a mai magyar elemző lexikonából hiányoztak. Sok esetben a tő ugyan megvolt, de a régi szövegekben más szófajú (is) volt, mint ma, illetve bizonyos konstrukciókban másképp kell elemezni őket, mint a mai megfelelőjüket. Ilyen például a régi névutós szerkezetek egy része, amelyben a névutó a -nAk-os birtokos szerkezethez hasonló formában egyeztetve van az NP fejével, ebben a ragos névutó elemzése más, mint az azonos alakú, ma is létező inkorporált névmást tartalmazó alaké. Kiemelkedően sok munkát jelentett a névmási elemet tartalmazó egységek paradigmáinak szabályszerű leírása.

az	**lövésnek**	**miátta**	oly	nehezen	nem	volna,	gyermeke
a	**lövésnek**	**miatta**	oly	nehezen	nem	volna	gyermeke,
a	**lövés**	**miatt**	oly	nehéz	nem	van	gyermek
Det	**N.Dat**	**PP.PxS3**	Adj|Pro	Adj.Essmod	Adv	V.Cond.S3	N.PxS3

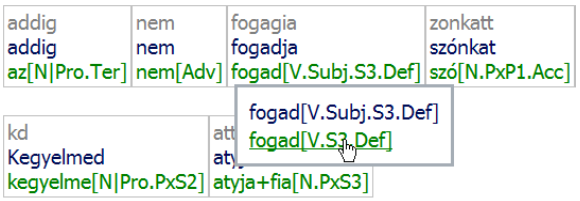
hogy	majd	megholt	**miatta.**
hogy	majd	meghalt	**miatta.**
hogy	majd	meg|+hal	**|+miatt**
C	Adv	VPfx.V.Past.S3	**PP.S3**

4 Egyértelműsítés

A néhány eleve elemzett formában meglévő szövegtől eltekintve a szövegek elemzését egyértelműsíteni is kellett. A lazább, megengedőbb elemző és a kibővített igei paradigmákban szereplő sok egybeesés, valamint a feljebb leírt eldönthetetlen többértelműségek ilyenként való címkézése miatt a történeti szövegekben a többértelműség aránya magasabb, mint a mai szövegek standard Humor elemzővel való elemzése esetében.

A morfoszintaktikai annotáció egyértelműsítésében a munka oroszlánrészét géppel végeztük. Az ó- és középmagyar elemző elemzéseit felhasználva eleinte a HMM-alapú HunPos taggert [1], később a PurePos taggert [8] inkrementális módon egyre több egyértelműsített és ellenőrzött szöveggel betanítva. Mivel a HunPos tövet nem ad vissza, csak címkét, a Humor elemzései közül a HunPos által választotthoz leghasonlóbb címkét tartalmazó elemzést választottuk. A PurePos esetében egyszerűbb a helyzet, mert ezt a feladatot saját hatáskörben elvégzi.

Az így egyértelműsített szövegek kézi ellenőrzéséhez (illetve az első szövegek még teljesen manuális egyértelműsítéséhez) olyan webes felületet hoztunk létre, amelyen a téves egyértelműsítések, illetve normalizálási hibák nagyon hatékonyan javíthatók. Az automatikusan választott elemzés helyett másikat az egérmutatót a szó fölé húzva automatikusan megjelenő listából választva lehet megadni. Kézzel is javítható akár az eredeti, akár a normalizált szóalak, akár az elemzés. A javítás után a szó azonnal újraelemeztethető, és új elemzés választható.



Az elemzőrendszert úgy alakítottuk ki, hogy alkalmas legyen arra, hogy a projekt során az alkalmazott annotáció egyes részleteit meg lehessen változtatni úgy, hogy ugyanakkor ne kelljen kidobni a korábban elvégzett egyértelműsítési munkát, hanem a korábban egyértelműsített szövegekbe is viszonylag egyszerűen átkerüljenek a módosított annotációk. Ennek alapjául az szolgál, hogy a szövegek újraelemzésekor a rendszer automatikusan a korábban megadott elemzéshez leghasonlóbb elemzést választja (az elemzésekből betűhármas-statisztikát készítve, és ezeket összehasonlítva). Bizonyos, az elemzőn végzett változtatások esetében (pl. amikor úgy döntöttünk, hogy a képzett igei alakoknak a korábbiaknál részletesebb elemzését használjuk) ennél kifinomultabb mechanizmusra volt szükség: a már meglevő egyértelműsített elemzéseket géppel generált reguláris kifejezésekkel konvertáltuk.

5 Keresés a korpuszban

A szövegekben való keresést támogató korpuszkezelő nemcsak azt teszi lehetővé, hogy különböző grammatikai szerkezetekre keressünk a szövegekben példákat, hanem azt is, hogy a kereső találataiban is azonnal kijavíthassuk az esetlegesen még az annotációban vagy a szövegben maradt hibákat, amely javítások ilyenkor az adatbázisba azonnal visszakerülnek. (A kereső utóbbi változata csak a megfelelő szakértelemmel és jogosultságokkal rendelkező felhasználók számára elérhető.) A hibakeresés és –javítás egyik hatékony módja, amikor a korpuszban kifejezetten olyan szerkezeteket keresünk, amelyek valószínűleg hibásak, és a valóban hibás találatokat azonnal javítjuk. A javított korpuszt ezután exportálni lehet, és a taggert a javított korpusszal újratanítani.

A keresőrendszer által használt korpuszadatbázis az Emdros korpuszkezelőn alapul [9]. A középmagyar korpusz lekérdezésére használható keresőben az Emdros eredeti lekérdezőszintaxisának (MQL) megfelelően megfogalmazott kérdések mellett egy az MQL-nél jóval tömörebb lekérdezőnyelv is használható. Az utóbbi formában megfogalmazott keresőkérdéseket a rendszer automatikusan MQL-re fordítja.

A kereső lehetővé teszi, hogy mondaton, tagmondaton, vagy adott metaadatokkal megjelölt tulajdonságú szövegen belül keressünk, illetve akár többmondatos egységek is lekérdezhetők. A kereső által megjelenített találati egység a mondat. A tagmondatok lehetnek nem folytonosak (ez az alárendelő szerkezetek esetén gyakran előfordul, de olykor a főmondat vagy egy mellérendelő szerkezet valamelyik eleme ékelődik be). Az alábbi példa olyan találati mondatot mutat be, amelyben több megszakított tagmondat is szerepel.

6 Összefoglalás

Cikkünkben egy ó- és középmagyar szövegek elemzésére is használható számítógépes morfológia kifejlesztésének legfontosabb lépéseit és az eközben felmerülő problémákat és megoldásukat mutattuk be. Emellett bemutattuk azt a keresőrendszert is, amely

lehetővé teszi az annotált szövegekben való keresés mellett azt is, hogy a keresés során kiderülő hibákat az erre jogosult felhasználók azonnal javítsák.

Amellett, hogy sikerült egy megbízhatóan működő, könnyen javítható elemzőprogramot és ennek felhasználásával morfológiailag elemzett történeti korpuszokat létrehozni, a projekt más tanulságokkal is bírt. A bA~bAn végződések speciális kódolása lehetővé tette, hogy a rag ingadozó helyesírásának változásáról számot adjunk. A történeti távlatokban létező szintaktikai többértelműségek néhány körét sikerült jól meghatározni és ezek kódolására, s ezáltal detektálására is sikerült módszert találnunk.

Az elkészült elemzővel a folyamatosan bővített ómagyar és középmagyar korpuszt elemezzük. Az elemzett adatbázisok kereshető formában részben már elérhetők. Az ómagyar korpusz itt: http://rmk.nytud.hu, a középmagyar korpusz feldolgozott része pedig ezen a címen: http://clara.nytud.hu/tmk.

Hivatkozások

1. Halácsy, P., Kornai, A, Oravecz, Cs.: HunPos: an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (2007) 209–212
2. Jakab L.. A Jókai-kódex mint nyelvi emlék szótárszerű feldolgozásban (Számítógépes nyelvtörténeti adattár 10.). Debreceni Egyetem, Debrecen (2002)
3. Jakab L., Kiss A.: A Guary-kódex ábécérendes adattára (Számítógépes nyelvtörténeti adattár 6.). Debreceni Egyetem, Debrecen (1994)
4. Jakab L., Kiss A.: Az Apor-kódex ábécérendes adattára (Számítógépes nyelvtörténeti adattár 7.). Debreceni Egyetem, Debrecen (1997)
5. Jakab L., Kiss A.: A Festetics-kódex ábécérendes adattára (Számítógépes nyelvtörténeti adattár 9.), Debreceni Egyetem, Debrecen (2001)
6. Németh M.: Nyelvi változás és váltakozás a műveltségi tényezők tükrében. Nyelvi változók a XVIII. században. Szegedi Tudományegyetem, Szeged (2008)
7. Novák A.: Milyen a jó humor? In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). Szegedi Tudományegyetem, Szeged (2003) 138–145
8. Orosz, Gy., Novák, A.: PurePos – an open source morphological disambiguator. In: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science. Wrocław, Poland (2012)
9. Petersen, U.: Emdros – A Text Database Engine for Analyzed or Annotated Text. In: Proceedings of the 20th International Conference on Computational Linguistics, Volume II. Geneva (2004) 1190–1193

Domének közti hasonlóságok és különbségek a szófajok és szintaktikai viszonyok eloszlásában

Vincze Veronika[1,2]

[1] MTA-SZTE Mesterséges Intelligencia Kutatócsoport
[2] Universität Trier, Linguistische Datenverarbeitung
vinczev@inf.u-szeged.hu

Kivonat: Ebben a cikkben a szófajok és szintaktikai relációk eloszlását vizsgáljuk különböző doménekben. A vizsgálat alapjául a Szeged Dependencia Treebank szolgál. Eredményeink alapján a szövegek témája (doménje) befolyásolja a szófajok, illetőleg a szövegszavak közti szintaktikai relációk eloszlását, így a domének között hasonlóságok és különbségek figyelhetők meg e téren, ami jelentőséggel bír különféle számítógépes nyelvészeti alkalmazásokban is, például a szófaji egyértelműsítők és a dependenciaelemzők hatékonyságának növelésében.

1 Bevezetés

A különféle nyelvi jelenségek eloszlásának kvantitatív vizsgálata nagy figyelmet kapott az utóbbi években. Számos nyelvben vizsgálták a szófajok és morfológiai jegyek eloszlását, l. például [3,9,12,13,14], mindemellett a szintaktikai viszonyok eloszlását is elemezték a kvantitatív szintaxis mint elmélet keretein belül [4,5,6,7,8].

A ragozó nyelvekben általában megfigyelhető a morfológia és a szintaxis szoros összefonódása, hiszen a nyelvtani relációk nagy részét morfológiai eszközök segítségével lehet kifejezni. Így e nyelvek kitűnő táptalajt biztosítanak a kvantitatív morfológiai és szintaktikai vizsgálatok számára. Például Köhler [6] a Szeged Treebank egy részén vizsgálta a nyelvtani viszonyok eloszlását, Väyrynen, Noponen és Seppänen [10] pedig a finnben elemzik a szemantikai viszonyokat.

Ebben a munkában a szófajok és szintaktikai viszonyok eloszlását vizsgáljuk különböző doménekhez tartozó magyar szövegekben. A vizsgálat alapjául a Szeged Dependencia Treebank [11] szolgál, amely hat különböző tématerületről tartalmaz kézzel annotált szövegeket: üzleti rövidhírek, újságcikkek, iskolai fogalmazások, szépirodalom, jogi és számítógépes szövegek. Kiinduló feltételezésünk szerint a szövegek témája (doménje) befolyásolja a szófajok, illetőleg a szövegszavak közti szintaktikai relációk eloszlását, így a domének között hasonlóságok és különbségek figyelhetők meg e téren, ami jelentőséggel bír különféle számítógépes nyelvészeti alkalmazásokban is, többek között a szófaji egyértelműsítők és a dependenciaelemzők hatékonyságának növelésében.

Vizsgálataink során az alábbi kérdésekre keressük a választ:

- Milyen jellemző eloszlási minták találhatók a magyar nyelvben a szófajokra és a szintaktikai viszonyokra nézve?
- A fenti eloszlások mennyire tekinthetők doménfüggőnek, illetve általánosnak?

A statisztikai adatok bemutatása és értelmezése mellett az eredmények nyelvészeti indoklására is törekszünk a cikkben.

2 A vizsgált korpusz

Vizsgálataink alapjául a Szeged Dependencia Treebank [11] szolgál. 82 000 mondatot, 1,5 millió szövegszót és 230 000 írásjelet tartalmaz hat doménből (iskolai fogalmazások, számítógépes szövegek, irodalom, jogi szövegek, újságcikkek és üzleti rövidhírek). A korpusz kézzel ellenőrzött morfológiai és szófaji, valamint szintaktikai (függőségi) elemzést is tartalmaz. A korpusz adatait az 1. táblázat foglalja össze.

1. táblázat: A Szeged Dependencia Treebank adatai.

	iskolás	számítógép	irodalom	jog	újság	rövidhír	összesen
Mondat	24 720	9 627	18 558	9 278	10 210	9 574	81 967
Írásjel	59 419	31 241	47 990	33 515	32 880	25 712	230 757
Szövegszó	283 591	183 562	189 751	225 207	190 406	201 527	1 504 801
Átlagos mondathossz	11,472	19,067	10,225	24,273	18,649	21,049	18,359

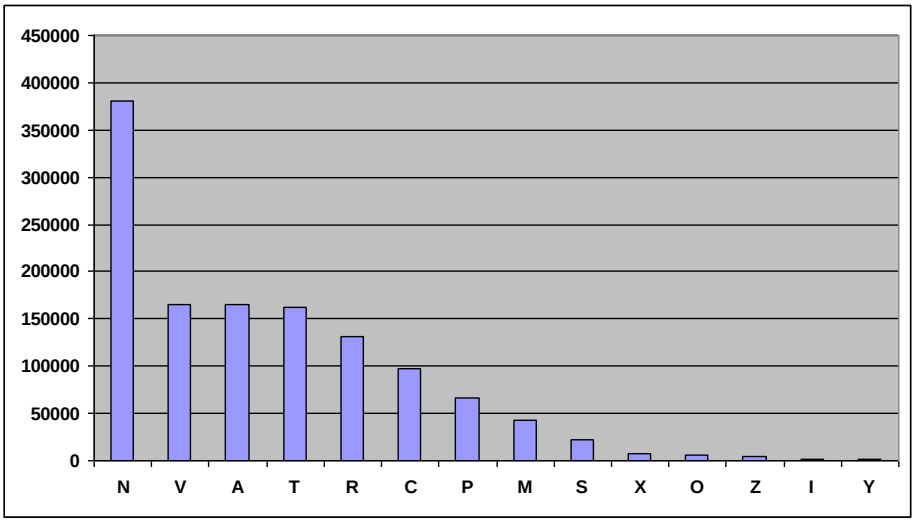
A következőkben a szófajok és függőségi viszonyok eloszlását vizsgáljuk meg a korpuszban és a különböző doménekben.

3 A szófajok eloszlása

A Szeged Dependencia Treebank az MSD morfológiai kódrendszert [2] használja a szófajok kódolására. Segítségével lehetőség nyílik mind a szófajok, mind a morfológiai jellemzők (például idő, mód, szám, személy, esetrag stb.) kódolására. Ebben a munkában kizárólag a fő szófaji információkra összpontosítunk, tehát minden token esetében csak a fő szófajt (főnév, ige, melléknév stb.) vesszük figyelembe, a finomabb morfológiai megkülönböztetésektől most eltekintünk, illetve az írásjeleket sem vonjuk be vizsgálataink körébe.

3.1 A szófajok eloszlása a teljes korpuszban

Az 1. ábrán látható a szófajok teljes korpuszbeli eloszlása. Az x tengely mutatja a szófajokat, az y tengelyen pedig a gyakorisági értékek láthatók.



1. ábra: A szófajok eloszlása a Szeged Dependencia Treebankben.

Amint az ábra is mutatja, a leggyakrabban előforduló szófajok a főnév, ige és melléknév. Ez összhangban van azzal az elvárással, hogy a szemantikai jelentéssel bíró lexikális elemek a leggyakoribbak. A névelők szintén gyakoriak, ami feltehetőleg annak köszönhető, hogy a főnevek igen gyakran szerepelnek névelő kíséretében a treebankben. Az ismeretlen szavak, rövidítések, helyesírási hibás szóalakok és a nyílt tokenosztályba tartozó szavak (X, Y, Z és O kódok) viszonylag ritkán fordulnak elő a korpuszban: összesítve a szavak 5,89%-át alkotják.

3.2 A szófajok eloszlása az egyes doménekben

A szófajok doménenkénti eloszlása a 2. táblázatban látható. A domének közti hasonlóságok részletesebb vizsgálatához felállítottuk az egyes szófajok gyakorisági rangsorát is, melyet a 3. táblázat szemléltet.

A szófajok eloszlásának vizsgálatához, illetve a domének közti hasonlóságok és különbségek megállapításához a Kendall-együtthatót (W) alkalmaztuk, amely a vizsgált elemek, jelen esetben a szófajok gyakorisági rangsorát felállítva mutatja meg, mennyire homogének a vizsgált szövegek. A Kendall-együttható értéke alapján a szövegek homogének (W = 0.9248), az eredmények szignifikánsak (DF=13, χ^2 = 154.571429). Azonban különbségek is megfigyelhetők az egyes domének között: míg a főnevek és igék az első két helyen szerepelnek az iskolás és az irodalmi szövegekben, addig a többi doménen nagy különbségek figyelhetők meg, lévén a főnév a leggyakoribb szófaj, ám az ige csak a negyedik-ötödik a gyakorisági rangsorban. A határozószavak viszonylag gyakran fordulnak elő az irodalmi és az iskolás szövegekben, ezzel szemben a melléknevek kevésbé gyakoriak, különösképpen a többi doménnel összevetve, ahol is a második vagy harmadik leggyakoribb szófajnak tekinthetők.

2. táblázat: A szófajok eloszlása doménenként.

	iskolás	irodalom	jog	újság	rövidhír	számítógép	összesen
főnév	56106	44737	78546	61902	79591	60201	381083
ige	58702	34805	15557	20751	16913	18958	165686
melléknév	20500	18403	40698	26955	32124	25887	164567
névelő	31253	19793	31495	25196	29027	26160	162924
határozószó	47322	29233	12725	17988	9760	14934	131962
kötőszó	29322	17348	15854	13695	7135	13522	96876
névmás	21081	14516	9549	8916	3620	9072	66754
számnév	7000	2374	6859	7032	14556	4817	42638
névutó	3286	2487	4268	3593	4933	2928	21495
ismeretlen	1026	1532	871	659	1297	2066	7451
nyílt tokenosztály	150	40	3663	284	779	827	5743
helyesírási hibás	2470	398	515	135	336	156	4010
indulatszó	738	814	6	135	5	114	1812
rövidítés	304	141	885	35	8	90	1463

3. táblázat: A szófajok gyakorisági rangsora doménenként.

	iskolás	irodalom	jog	újság	rövidhír	számítógép
ige	1	2	5	4	4	4
főnév	2	1	1	1	1	1
határozószó	3	3	6	5	6	5
névelő	4	4	3	3	3	2
kötőszó	5	6	4	6	7	6
névmás	6	7	7	7	9	7
melléknév	7	5	2	2	2	3
számnév	8	9	8	8	5	8
névutó	9	8	9	9	8	9
helyesírási hibás	10	12	13	12	12	12
ismeretlen	11	10	12	10	10	10
indulatszó	12	11	14	13	14	13
rövidítés	13	13	11	14	13	14
nyílt tokenosztály	14	14	10	11	11	11

Az egyes domének közti hasonlóságok és különbségek további vizsgálatához minden egyes doménpárra kiszámoltuk a Kendall-együttható értékét. Az eredményeket a 4. táblázat mutatja (minden eredmény szignifikáns).

4. táblázat: A domének hasonlósága a szófajok eloszlása terén.

	újság	rövidhír	számítógép	irodalom	iskolás	jog
újság		0,9802	0,9978	0,9626	0,9363	0,9758
rövidhír	0,9802		0,9780	0,9319	0,9055	0,9626
számítógép	0,9978	0,9780		0,9648	0,9429	0,9736
irodalom	0,9626	0,9319	0,9648		0,9824	0,9253
iskolás	0,9363	0,9055	0,9429	0,9824		0,9033
jog	0,9758	0,9626	0,9736	0,9253	0,9033	

A fenti eredmények alapján a szófajok eloszlása terén a két leghasonlóbb domén az újságcikkek és a számítógépes szövegek (W = 0.9978). A hasonlóságot magyarázhatja az a tény, hogy a számítógépes szövegek egy része valójában egy számítógépes magazinból származik, így egyaránt rendelkezik a számítógépes szövegekre, illetve az újságcikkekre jellemző sajátosságokkal. Az iskolai fogalmazások a szépirodalmi szövegekhez hasonlítanak leginkább (W = 0.9824), azonban eltérnek az üzleti hírektől és a jogi szövegektől. A legnagyobb különbséget a jogi és iskolás szövegek között figyelhetjük meg, amely feltehetőleg a domének között húzódó alapvető stilisztikai különbségeknek köszönhető. Az iskolai fogalmazásokban a mondatok jóval rövidebbek, továbbá többnyire a szerzőjükkel megtörtént eseményeket írnak le, az események leírása pedig az igék fokozottabb használatát követeli meg. Ezzel szemben a jogi szövegek nem annyira eseményeket, hanem inkább tényeket és állapotokat írnak le, így kevesebb igét is tartalmaznak.

4 A szintaktikai viszonyok eloszlása

A Szeged Treebank 2.0-ban található igék és bővítményeik közti szintaktikai viszonyokat először automatikusa konvertálták függőségi viszonyokra [1], majd ezeket kézzel ellenőrizve és javítva állt elő a Szeged Dependencia Treebank [11]. A függőségi viszonyok eloszlását a szófajokéhoz hasonlóan elemezzük a következőkben.

4.1 A szintaktikai viszonyok eloszlása a korpuszban

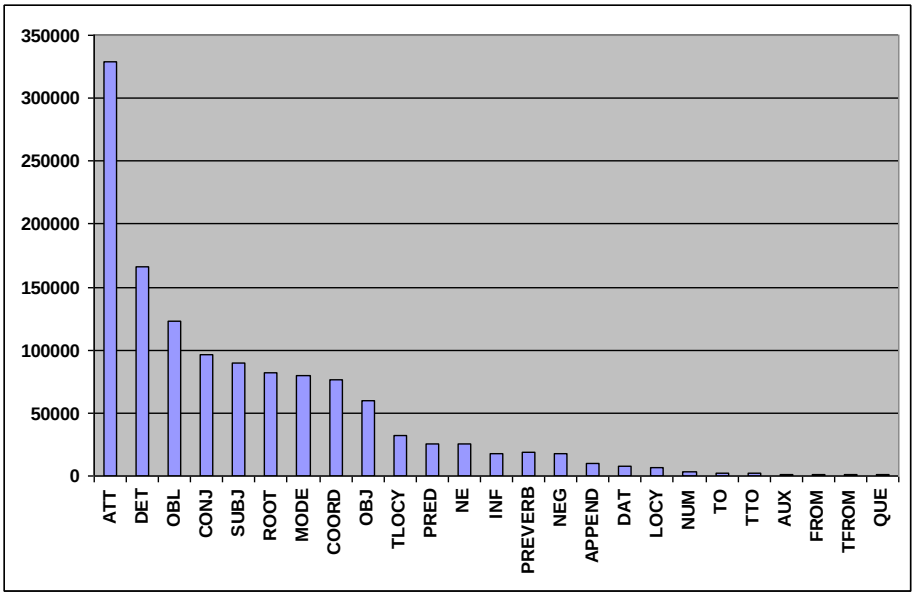
A szintaktikai viszonyok eloszlásának tanulmányozásához először is megszámoltuk, milyen viszonyok húzódnak az egyes szavak között a korpuszban (az írásjeleket ismét figyelmen kívül hagyva). Az eredményeket az 5. táblázat mutatja.

5. táblázat: A szintaktikai viszonyok eloszlása doménenként.

	iskolás	irod.	jog	újság	rövidhír	sz.gép	összesen	%
ATT	46046	35386	80603	51891	66889	48221	329036	25,82
DET	32044	20477	32187	25536	29192	26618	166054	13,03
OBL	23502	15771	23182	18044	25143	17168	122810	9,64
CONJ	29236	17108	15584	13569	7114	13296	95907	7,53
SUBJ	20650	15784	12094	14144	14181	12615	89468	7,02
ROOT	24723	18564	9284	10210	9577	9658	82016	6,44
MODE	24253	15320	11051	11432	7533	10345	79934	6,27
COORD	20553	12852	11533	11600	7959	12073	76570	6,01
OBJ	14077	9547	9609	9553	7180	9433	59399	4,66
TLOCY	12533	5983	2257	4349	3854	2897	31873	2,50
PRED	8014	5045	3655	3348	1696	3949	25707	2,02
NE	764	1187	1344	4535	11820	5447	25097	1,97
INF	7847	2796	2437	1877	618	2513	18088	1,42
PREVERB	4357	3214	2698	2620	2668	2719	18276	1,43
NEG	5734	4009	2788	2406	862	1722	17521	1,38
APPEND	1157	913	2769	1248	1188	2156	9431	0,74
DAT	2259	1401	1365	1397	759	1108	8289	0,65
LOCY	2912	2041	237	779	328	616	6913	0,54
NUM	5	0	99	593	2277	365	3339	0,26
TO	1198	764	66	278	245	142	2693	0,21
TTO	654	379	130	315	166	177	1821	0,14
AUX	318	476	36	146	25	53	1054	0,08
FROM	285	262	91	165	32	126	961	0,08
TFROM	213	243	13	214	197	84	964	0,08
QUE	202	209	106	155	20	104	796	0,06

A függőségi viszonyok eloszlása a teljes korpuszon a 2. ábrán látható. Amint láthatjuk, a leggyakoribb reláció az ATT, mely az összes szintaktikai viszony körülbelül negyedét adja. Az ATT általános módosító viszonynak tekinthető, melybe a jelzői és alárendelői szerepek egyaránt beletartoznak, vagyis szavakat és tagmondatokat egyaránt összekapcsolhat. Ennek a ténynek feltehetőleg fontos szerepe van a reláció gyakori előfordulásában. Mivel a főnevek általában egy névelővel együtt fordulnak elő, a DET reláció is viszonylag gyakran szerepel a szövegekben (13%). A harmadik leggyakoribb szintaktikai viszony, az OBL számos magyar esetragot foglal magába, emiatt a gyakorisága is megfelel ezen esetragok összesített gyakoriságának. A kötőszavak

is viszonylag gyakoriak a korpuszban, ami azt sugallja, hogy számos alá- és mellérendelés található a korpuszban, mind tagmondatok, mind szavak szintjén.



2. ábra: A függőségi viszonyok eloszlása a Szeged Dependencia Treebankben.

4.2 A szintaktikai viszonyok eloszlása az egyes doménekben

Szerettük volna megvizsgálni azt is, hogy az egyes doménekre nézve milyen sajátosságok mutathatók ki a szintaktikai viszonyok eloszlására nézve, illetőleg milyen hasonlóságok és különbségek figyelhetők meg a domének között. A függőségi viszonyok rangsorát a 6. táblázat szemlélteti.

A táblázatbeli adatok alapján szintén kiszámoltuk a Kendall-együttható értékét a korpuszra, és azt találtuk, hogy az adatok homogének (W= 0.9321). Az eredmények szignifikánsak (DF=25, $\chi^2 = 134.221538$).

A domének közti összehasonlítás számos érdekességet tartogat. Először is az üzleti rövidhírek bővelkednek a több tagból álló tulajdonnevekben és számnevekben, hiszen számos, cégekkel kapcsolatos pénzügyi hírt tartalmaznak. Így az NE és NUM relációk is igen gyakran fordulnak elő ezen a doménen. Másodszor, a jogi szövegekben igen nagy mennyiségben fordul elő az APPEND reláció, mely a mondatba szorosan nem tartozó közbevetéseket jelzi. A jogi szövegekben számos utalás található törvényekre, paragrafusokra, melyek nem képezik a mondat szerves részét, így az APPEND relációval kapcsolódnak a többi elemhez.

6. táblázat: A szintaktikai viszonyok gyakorisági rangsora doménenként.

	iskolás	irodalom	jog	újság	rövidhír	számítógép
ATT	1	1	1	1	1	1
DET	2	2	2	2	2	2
CONJ	3	4	4	5	10	4
ROOT	4	3	9	8	6	8
MODE	5	7	7	7	8	7
OBL	6	6	3	3	3	3
SUBJ	7	5	5	4	4	5
COORD	8	8	6	6	7	6
OBJ	9	9	8	9	9	9
TLOCY	10	10	15	11	11	12
PRED	11	11	10	12	14	11
INF	12	14	14	15	18	14
NEG	13	12	11	14	16	16
PREVERB	14	13	13	13	12	13
LOCY	15	15	18	18	19	18
DAT	16	16	16	16	17	17
TO	17	19	23	21	20	21
APPEND	18	18	12	17	15	15
NE	19	17	17	10	5	10
TTO	20	21	19	20	22	20
AUX	21	20	24	25	24	25
FROM	22	22	22	23	23	22
TFROM	23	23	25	22	21	24
QUE	24	24	20	24	25	23
NUM	25	25	21	19	13	19

A függőségi viszonyok esetében szintén a Kendall-együtthatót alkalmaztuk a domének közti hasonlóságok felderítésére, az eredmények ez esetben is szignifikánsak. A 7. táblázatban látható eredmények alapján az egymáshoz leghasonlóbb doménpárok az újságcikkek és a számítógépes szövegek, illetőleg a fogalmazások és a szépirodalmi szövegek (W = 0.9965 és 0.995, rendre). A legnagyobb eltérés pedig a fogalmazások és az üzleti rövidhírek között mutatkozik (W = 0.8973), hasonlóan a szófajok eloszlásához, ami a stilisztikai eltérésekre vezethető vissza.

7. táblázat: A domének hasonlósága a szintaktikai viszonyok eloszlása terén.

	újság	rövidhír	számítógép	irodalom	iskolás	jog
újság		0,9762	0,9962	0,9665	0,9577	0,9723
rövidhír	0,9762		0,9708	0,9158	0,8973	0,9227
számítógép	0,9962	0,9708		0,9627	0,9565	0,9777
irodalom	0,9665	0,9158	0,9627		0,995	0,9627
iskolás	0,9577	0,8973	0,9565	0,995		0,9588
jog	0,9723	0,9227	0,9777	0,9627	0,9588	

5 Az eredmények értelmezése

Az eredmények alapján kirajzolódnak az alkorpuszok (illetve domének) közti hasonlóságok, illetve távolságok. Mind a szófajok, mind a szintaktikai viszonyok szempontjából a legnagyobb hasonlóságot az újságcikkek és a számítógépes szövegek mutatták. A hasonlóságot magyarázhatja, hogy a számítógépes szövegek jó része valójában egy számítástechnikai témájú magazinból származik, így a nyelvezetük erősen hasonlít a sajtónyelvre, csakúgy, mint az újságcikkek nyelvezete. A szépirodalmi szövegek és az iskolai fogalmazások közti hasonlóság azzal magyarázható, hogy mindkét esetben történetek elbeszéléséről van szó, tehát az elbeszélő stílus jegyei figyelhetők meg mindkét domén szövegeiben. Az üzleti hírek, illetve a jogi szövegek pedig egyedi nyelvi jellemzőkkel bírnak.

6 Az eredmények alkalmazása a számítógépes nyelvészetben

A vizsgálat eredményei számos területen hasznosíthatók a számítógépes nyelvészetben. Mivel a magyar szófaji egyértelműsítők és szintaktikai elemzők nagy része a Szeged (Dependencia) Treebanket használja tanító adatbázisként (pl. [15]), a domének közti különbségek jelentősen befolyásolhatják azt, hogy mely részkorpuszokat érdemes tanító adatbázisként kiválasztani egy adott elemzendő szöveghez. Például egy regény szintaktikai elemzésekor valószínűleg az iskolai fogalmazások és a szépirodalmi szövegek unióján tanított elemző éri el a legjobb eredményt. A domének közti hasonlóságoknak és különbségeknek a részletes elemzése megnyitja az utat a különféle doménadaptációs technikáknak a szófaji egyértelműsítésben és szintaktikai elemzésben való alkalmazása előtt is. Végül az eloszlási minták, pontosabban a domének közti hasonlóságok és különbségek elemzése a dokumentumosztályozásban is hasznosítható.

A fentiek alátámasztására végeztünk egy kísérletet. Az *Egri csillagok* című regényt morfológiailag elemeztük, szófajilag egyértelműsítettük, majd dependenciaelemzésnek vetettük alá a *magyarlanc* elemzővel [15]. Az elemzések alapján meghatároztuk a szófajok és a szintaktikai viszonyok eloszlását, majd az eloszlási mintákat összevetet-

tük a Szeged Dependencia Treebank minden egyes doménjével, és minden párra kiszámítottuk a Kendall-együtthatót. A szignifikáns eredmények a 8. táblázatban láthatók.

8. táblázat: Az *Egri csillagok* és a domének hasonlósága a szófajok és a szintaktikai viszonyok eloszlása terén.

	szófaj	dependencia
iskolás	0,9868	0,9865
irodalom	0,9934	0,9946
újság	0,9604	0,9638
számítógép	0,9670	0,9546
rövidhír	0,9341	0,9212
jog	0,9297	0,9527

Az eredmények azt mutatják, hogy mind a szófajok eloszlása, mind a függőségi viszonyok eloszlása terén az *Egri csillagok* a legnagyobb fokú hasonlóságot az irodalmi szövegekkel mutatja, illetve a második leginkább hasonló domén az iskolás szövegek. Vagyis ha nem tudnánk a szöveg műfaját, akkor is az eredmények alapján a legnagyobb valószínűséggel irodalmi szövegnek titulálnánk egy dokumentumosztályozási feladat során. Mivel tudjuk, hogy az *Egri csillagok* is az irodalmi művek sorába tartozik, így ez a döntés helytálló lenne. A fenti számok ugyanakkor megerősítik azt a korábbi eredményünket is, miszerint az iskolás és az irodalmi szövegek hasonlítanak egymáshoz.

7 Összegzés

A cikkben a szófajok és szintaktikai relációk eloszlását vizsgáltuk különböző doménekben. A vizsgálat alapjául a Szeged Dependencia Treebank szolgált. Eredményeink alapján a szövegek doménje befolyásolja a szófajok, illetőleg a szövegszavak közti szintaktikai relációk eloszlását, így a domének között hasonlóságok és különbségek figyelhetők meg e téren. Mind a szófajok, mind a szintaktikai viszonyok szempontjából a legnagyobb hasonlóságot az újságcikkek és a számítógépes szövegek mutatták. Az irodalmi és az iskolás szövegek szintén hasonlítanak egymásra, az üzleti hírek és a jogi szövegek pedig önálló sajátságokkal bírnak.

A jövőben szeretnénk megvizsgálni a szófajok és a szintaktikai jellemzők eloszlását más szövegtípusokban is, illetőleg a fenti eredmények felhasználásával szeretnénk doménadaptációs kísérleteket végezni a szófaji egyértelműsítés és dependenciaelemzés területén.

Köszönetnyilvánítás

Szeretnék köszönetet mondani Reinhard Köhlernek a munkámat segítő számos hasznos tanácsáért és értékes megjegyzéséért.

A kutatás – részben – az A/11/83421 jelű fiatal kutatói ösztöndíj keretében a Deutscher Akademischer Austauschdienst támogatásával, illetve a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt keretében az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett valósult meg.

Hivatkozások

1. Alexin Z.: A frázisstrukturált Szeged Treebank átalakítása függőségi fa formátumra. In: V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2007). Szegedi Tudományegyetem, Szeged (2007) 263–266
2. Alexin, Z., Csirik, J., Gyimóthy, T., Bibok, K., Hatvani, Cs., Prószéky, G., Tihanyi, L.: Annotated Hungarian National Corpus. In: Proceedings of EACL (2003) 53–56
3. Best, K.-H.: Word class frequencies in contemporary German short prose texts. Journal of Quantitative Linguistics, Vol. 1 (1994) 144–147
4. Cech, R., Pajas, P., Mačutek, J.:. Full valency. Verb valency without distinguishing complements and adjuncts. Journal of Quantitative Linguistics, Vol. 17, No. 4 (2010) 291–302
5. Köhler, R.: Syntactic Structures. Properties and Interrelations. Journal of Quantitative Linguistics, Vol. 6, No. 1 (1999) 46–57
6. Köhler, R.: Quantitative Syntax Analysis. de Gruyter, Berlin, New York (2012)
7. Liu, H.: Probability distribution of dependency distance. Glottometrics, Vol. 15 (2007) 1–12
8. Liu, H.: Probability distribution of dependencies based on Chinese dependency treebank. Journal of Quantitative Linguistics, Vol. 16, No. 3 (2009) 256–273
9. Tuzzi, A., Popescu, I.-I., Altmann, G.: Quantitative analysis of Italian texts. RAM, Lüdenscheid (2010)
10. Väyrynen, P. A., Noponen, K., Seppänen, T.: Preliminaries to Finnish word prediction. Glottotheory, Vol. 1 (2008) 65–73
11. Vincze, V. Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (2010)
12. Vulanović, R., Köhler, R.: Word order, marking, and Parts-of-Speech Systems. Journal of Quantitative Linguistics, Vol. 16, No. 4 (2009) 289–306
13. Ziegler, A.: Word class frequencies in Brazilian-Portuguese press texts. Journal of Quantitative Linguistics, Vol. 5 (1998) 269–280
14. Ziegler, A.: Word class frequencies in Portuguese press texts. In: Uhlířová, L., Wimmer, G., Altmann, G., Köhler, R. (eds.): Text as a linguistic paradigm: levels, constituents, constructs. Festschrift in honour of Luděk Hřebíček. Wissenschaftlicher Verlag Trier, Trier (2001) 295–312
15. Zsibrita J., Vincze V., Farkas R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 368–374

Gondolatok a (magyar) statisztikai szintaktikai elemzőkről

Farkas Richárd

Szegedi Tudományegyetem, Informatikai Tanszékcsoport
rfarkas@inf.u-szeged.hu

Kivonat: Jelen munkában áttekintést adunk a statisztikai szintaktikai elemzés nemzetközi állapotáról, a magyar szintaktikai elemzők fejlesztéséhez hasznos szempontok szem előtt tartásával. Négy kérdéscsoportot tárgyalunk bővebben: (i) Mi a különbség és hasonlóság különböző nyelvek szintaktikai elemzése/elemezhetősége közt? Érvelünk amellett, hogy a magyar nyelvre kidolgozott elemzési módszerek más nyelvek elemzéséhez is hasznos tanulságokkal szolgálhatnak. (ii) Tárgyaljuk, hogy a magyar nyelv statisztikai szintaktikai elemzése nem mondható nehezebbnek, mint bármely más nyelvé, de az elemzők továbbfejlesztéshez nyelvspecifikus módszerek kidolgozása szükséges. (iii) Általánosságban összevetjük továbbá a két legelterjedtebb szintaktikai reprezentációt, a konstituens- és függőségi reprezentációkat és (iv) érvelünk a belső kiértékelési metrikák kizárólagos használata ellen.

1 Bevezetés

A szintaxis a szavak szószerkezetekké és mondatokká kapcsolódásának szabályait írja le. Az egyes mondatok szintaktikai elemzése igen fontos bemenete számos számítógépes nyelvészeti alkalmazásnak, mint például információkinyerés, véleménydetekció, kivonatolás vagy gépi fordítás.

Jelen munkában rövid áttekintést adunk a statisztikai szintaktikai elemzés nemzetközi állapotáról, majd részletesen tárgyalunk négy témát, amelyek – véleményünk szerint – a közeljövő statisztikai szintaktikai kutatásait uralni fogják és a magyar szintaktikai elemzők fejlesztésének szempontjából is alapvető fontosságúak.

Statisztikai elemző alatt *adatvezérelt* (data-driven) elemzőket értünk, azaz olyan megközelítéseket, ahol a nyelvtani mintázatok egy kézzel annotált korpusz (adat) formájában állnak rendelkezésre és a cél olyan elemző készítése, amely a korpusz elemzéseit próbálja automatikusan reprodukálni. Jelen munkában kizárólag ezekre koncentrálunk és nem célunk, hogy ezeket a kézzel írt nyelvtanokkal összehasonlítsa. A tanulmány aktualitását az adja, hogy míg a nemzetközi porondon a statisztikai szintaktikai elemzők túlnyomó többségben vannak, addig magyarra csak néhány ilyen kezdeményezést ismerünk. Reményeink szerint ez a tanulmány hozzájárul statisztikai technikák kiaknázásához, magyar nyelvre történő adaptálásához vagy kidolgozásához.

2 Miért érdekes a magyar nyelv szintaktikai elemzése a nemzetközi kutatásban?

Az elmúlt két évtizedben az angol szintaktikai elemzők látványosan fejlődtek, azonban szinte minden statisztikai szintaktikai elemző módszer angol nyelvre lett kidolgozva. Ugyan az elmúlt években erősödő trend, hogy egyéb nyelvek szintaktikai elemzését is vizsgáljuk, az angolra kidolgozott módszerek adaptációja közel sem triviális, gondoljunk csak olyan nyelvekre, ahol a szavak összetett belső szerkezettel rendelkeznek vagy a szórend szabad [1].

A természetes nyelveket szintaktikai elemzés szempontjából egy ún. konfigurációs tengelyre helyezhetjük fel. A spektrum egyik végén az angol mint erősen konfigurációs nyelv található, míg a másik végén a magyar (a tagalog és warlpiri mellett), ahol a legtöbb mondatszintű szintaktikai információt a morfológia kódolja. De természetesen még a végleteken sem beszélhetünk tisztán konfigurációs vagy csak tisztán nemkonfigurációs nyelvekről, hiszen a morfológia az angolban is fontos szerepet játszik, és a magyarban is vannak szórendi megkötések.

A konfigurációs (vagy morfológiai gazdagsági) spektrum szempontjából az egyes nyelveket három szorosan kapcsolódó jelenség mentén érdemes vizsgálni. A *morfológiai gazdagság*ot mérhetjük azzal, hogy egy adott szónak hány morfológiai alakja lehetséges. Például egy főnévnek angolban 2, németben 8, míg magyarban több száz formája lehet. A *szinkretizmus* szintje mérhető azzal, hogy ugyanazon szóalak hány különféle morfológiai alaknak feleltethető meg. Végül a *konfigurációsság* szintje arra vonatkozik, hogy a szavak és frázisok sorrendjének mennyire erős szerepe van a szintaktikai kapcsolatok reprezentálásában. Az angol erősen konfigurációs nyelv, a szórend meghatározza az egyes főnévi csoportok nyelvtani szerepét, míg a magyarban szinte bármilyen sorrend lehet nyelvtanilag helyes [2].

A konfigurációsság és a morfológiai gazdagság közötti negatív korreláció nyilvánvaló. A gazdag morfológia jelöli a nyelvtani szerepeket, nem szükséges azokat még a szórenddel is kifejezni. Másrészről, ha a morfológia nem ad elég támpontot, akkor a konfigurációs nyelvek a szórend rögzítésével tudják az egyes nyelvtani szerepeket kifejezni. Például angolban az igét követő főnévi csoport a tárgy, így nem szükséges azt a morfológia szintjén is jelölni. A szinkretizmus egy köztes megoldásnak is tekinthető a gazdag morfológia és a szórend rögzítése között. Segítségével egy gazdagabb morfológia kevesebb – igaz, többértelmű – felszíni formával kifejezhető. A többértelműség pedig feloldható szórendi jelek alapján (amelyek a kötött szórendnél kevésbé szigorú szabályokat használnak).

A fenti gondolatmenet alapján a morfológiailag gazdag(abb) nyelvekre fejlesztett szintaktikai elemzők legnagyobb kihívása, hogy hogyan valósíthatnak meg az angol rendszereknél erősebb együttműködést a morfológiai elemzés és a szintaktikai elemzés közt. Nyitott kutatási kérdések [3], hogy

- Milyen morfológiai információkat érdemes felhasználni a szintaktikai elemzéshez?
- Hogyan érdemes a morfológiai információt reprezentálni (a szófaji kódok, frázisok, függőségi élek szintjén)?

- Hogyan hatnak egymásra a morfológiai és szintaktikai jelenségek és hogyan lehet ezek kölcsönhatását hatékonyan kiaknázni?
- Hogyan kezeljük az ismeretlen szóalakokat, amelyek nagyon gyakran csak egy ismert szó korábban nem látott morfológiai formája?

Ezeknek a kérdéseknek a vizsgálata kapcsán a magyar mint állatorvosi ló érdekes szerepet tölthet be a morfológiailag gazdag nyelvekre kidolgozott módszerek tesztelésében. Sőt érdekes tanulságokkal szolgálhatnak a konfigurációs spektrum közepén helyet foglaló nyelvek, mint például a német számára is [2].

3 Nehéz-e a magyar szintaktikai elemzés?

A szakmai közbeszédben gyakran hallunk olyan kijelentéseket, hogy egyik vagy másik nyelv szintaktikai elemzése „nehezebb" feladat, mint másiké. Ráadásul számítógépes nyelvészeti körökben ezt a statisztikai elemzők által elért pontosságnak szokták megfeleltetni. Például a magyarról a „CoNLL-2007 többnyelvű függőségi elemzés" verseny [4] óta az volt a közgondolkodás, hogy a magyar szintaktikai elemzés egy nagyon nehéz feladat, mivel a legjobb rendszerek közel 10 százalékponttal rosszabb eredményeket értek el a magyar korpuszon, mint az angolon.

Véleményünk szerint ezekből a számokból nem szabad messzemenő következtetéseket levonni. A pontosságmetrikák közvetlen összehasonlítása például azonnal megkérdőjelezhető, ha arra gondolunk, hogy egy angol mondat 20%-kal több szót tartalmaz, mint egy magyar, és ennek a többletnek (elöljárók, személyes névmás) az elemzése relatíve egyszerű.

A [5] munkában megmutattuk, hogy magyar függőségi korpuszon is elérhető az angolhoz közeli eredmény:

1. táblázat: State-of-the-art függőségi elemzés eredményei magyar és angol nyelven. „dev" és „test" két különböző kiértékelési alkorpusz. LAS (labeled attachment score): a token szülőjének és élcímkéjének is helyesnek kell lennie. ULA (unlabeled attachment score): az élcímkézés nem releváns. Az értékek zárójelben etalon szófaji kódok alkalmazása mellett.

		ULA	LAS
Szeged Dependencia Treebank	dev	89,7 (91,1)	86,8 (89,0)
	test	90,1 (91,5)	87,2 (89,4)
CoNLL-2009 angol korpusz	dev	91,6 (92,7)	88,5 (90,0)
	test	92,6 (93,4)	90,3 (91,5)

Ez annak tudható be, hogy magyarra a morfológiai elemző [6] és egyértelműsítő igen jó hatékonysági fokkal működik, és ahogyan azt az előző fejezetben is tárgyaltuk, a magyarban a morfológia kódolja a nyelvtani szerepek jelentős részét, így a szintaktikai elemzés viszonylag egyszerű feladatnak mondható.

Véleményünk szerint a statisztikai elemzők (mind konstituens, mind dependencia) mára elérték azt a fejlettségi szintet, hogy algoritmikus, nyelvfüggetlen javításokkal

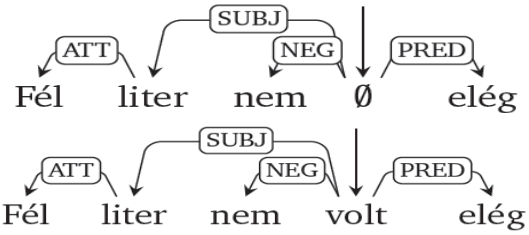
már jelentősen nem javíthatóak. Ehelyett az egyes nyelvek (és annotációs irányelvek) sajátosságait figyelembe vevő megoldások szükségesek. Az [5] munka keretében szisztematikusan elemeztük az angol és magyar függőségi elemzők hibáit és megmutattuk, hogy a hibák nagy része nyelvspecifikus.

3.1 Statisztikai elemzők a CoNLL-2007 és a Szeged Dependencia Treebankeken

Az [5] cikkben azt is tárgyaltuk, hogy a különbség a 2007-es és a 2012-es eredmények közt az annotáció különbségeinek tudható be. Egyrészt míg 2007-ben a frázisstruktúrákból automatikusan konvertált függőségi fák álltak csak rendelkezésre, 2011-re elkészült a Szeged Dependencia Treebank [7], amelyben az automatikus konverzió kimenetét manuálisan javították. A kézi javítás – elsősorban a mellérendelések újfajta kezelésének és a melléknévi frázisok belső szerkezetének köszönhetően – egy tisztább tanuló és kiértékelő adatbázist eredményezett. Másrészt maga az annotációs séma is megváltozott. Például a korábbi 49 élcímke helyett csak 29 szerepel a 2011-es korpuszban (elkerülve, hogy a nyelvtani szerepek duplán, az esetragokban és az élcímkéken is jelölve legyenek).

3.2 Függőségi elemzés virtuális csomópontokkal

Az [5] hibaelemzése szintén rávilágított arra, hogy a Szeged Dependencia Treebank virtuális csomópontjai okozzák a legtöbb problémát a statisztikai elemzőknek. A korpuszépítés során virtuális csomópontok kerültek beszúrásra abban az esetben, ha a létige (kijelentő mód jelen idő E/3. alak) nem jelenik meg a felszínen, illetve elliptikus összetételekben. A virtuális csomópontok az igét helyettesítik ezeken a helyeken. A virtuális csomópontok bevezetésének motivációja kettős. Egyrészt a függőségi elemzők számára (is) az ige a mondat központi eleme, így az ige nélküli mondatokkal nagyon nehezen boldogulnának. Másrészt a szintaktikai elemzést felhasználó célalkalmazások (például gépi fordítás) számára hasznos, ha például jelen és múlt idejű szerkezetek ugyanolyan struktúrában jelennek meg. Erre példa az alábbi két elemzési fa is:



Mivel a függőségi elemzés (fa) csomópontjai általában a mondat szavai, ezért az – angol nyelvet szem előtt tartva kidolgozott – elemzők nincsenek felkészítve arra, hogy új csomópontot legyenek képesek beszúrni az elemzés folyamán. A [8] munka keretében kidolgoztunk és összehasonlítottunk három eljárást a virtuális csomópontok automatikus beszúrására:

- előfeldolgozó: a nyers mondatba szúrunk be virtuális tokeneket, majd a sztenderd elemzőt alkalmazzuk. Ahhoz, hogy eldöntsük, hova érdemes beszúrni virtuális tokent, azonosítjuk a tagmondat-hierarchiát és lényegében igét nem tartalmazó tagmondatokat keresünk
- tranzakcióalapú elemző: felvettünk egy új átmenetet, ami képes új csomópontokat beszúrni
- virtuális csomópontok kódolása élcímkéken: itt a fát átalakítjuk úgy, hogy a virtuális csomópontok gyerekeinek a szülője a virtuális pont szülője lesz, élcímkéje pedig a két él címkéje összefűzve. Ezen a fán sztenderd elemzőket taníthatunk, majd azok kimenete alapján, az összetett címkék helyére virtuális csomópontot tudunk beszúrni.

A kísérleteket a Szeged Dependencia Treebanken és a német Tiger Treebank dependenciaverzióján végeztük el. Azt a következtetést vontuk le, hogy az előfeldolgozós módszer alulmarad a másik két módszerrel szemben, de az nem egyértelmű, hogy a kibővített tranzakcióalapú vagy az éleken kódolós módszer minden esetben jobb lenne a másiknál. A [8] eredményei azt is megmutatták, hogy a lokális jelenségek – mint a létige ki nem fejeződése – jó hatékonysággal megoldható problémák, míg azoknál az eseteknél (például ellipszis), ahol távoli függőségek azonosítása szükséges, a statisztikai módszerek igen alacsony pontosságot tudtak elérni.

Habár virtuális csomópontok beszúrása az angolban is szükséges lenne, ezek az esetek annyira ritkák, hogy nem foglalkoznak velük az elemzők. A virtuális csomópontok kérdése tehát egy jó példa arra, hogy (i) az angolcentrikus elemzőket nem lehet egyszerűen adaptálni egyéb nyelvekre, illetve (ii) hogy a magyar korpusz alapján kidolgozott megoldások hasznosak lehetnek egyéb nyelvek statisztikai elemzőinek kidolgozásához.

4 Függőségi vagy konstituensalapú elemző?

A létező számos szintaktikai reprezentáció közül a statisztikai módszerek túlnyomó többsége konstituens- vagy függőségi reprezentáción alapul. A kettő közül is az elmúlt 6-7 évben a függőségi elemzők lettek a divatos(abb)ak, annak ellenére, hogy semmi sem bizonyítja, hogy a függőségi elemzők jobbak vagy hasznosabbak lennének, mint a konstituenselemzők, mint ahogy azt sem, hogy kevésbé jók vagy hasznosak.

A függőségi reprezentáció előnyeként azt szokták felhozni, hogy abban a nemprojektív élek (nem folytonos konstituensek), illetve a nyelvtani szerepek egyszerűen ábrázolhatóak. Azonban ez nem vonja maga után, hogy az elemzők képesek is lennének ezt kielégítő pontossággal automatikusan reprodukálni. Például a legtöbb függőségi elemző első lépésben egy projektív elemzést generál, majd egy különálló második lépésben utófeldolgozva a fát kap nem projektív elemzéseket. Hasonló utófeldolgozási eljárás alkalmazható lenne konstituensfákon is [2]. Ráadásul vannak olyan nyelvi jelenségek is, amelyeknek viszont a konstituensreprezentáció a természetesebb módja. Ilyenek a mellérendelések, a tagmondatok hierarchikus viszonya és a frázishatárok.

Fontos megjegyeznünk, hogy ugyanakkor mindezen nyelvi jelenségek reprezentálhatóak mindkét megközelítésben [9].

A függőségi elemzők tényleges nagy előnye a sebességük. A szabadon elérhető függőségielemző-implementációk nagyságrendileg húszszor gyorsabbak, mint a konstituenselemzők. Ha az elemzők aszimptotikus időkomplexitását nézzük, mindkét reprezentációhoz léteznek lineáris idejű inkrementális elemzők. A gyakorlatban azonban a nyelvtan mérete miatt a keresési tér sokszorosa a konstituenselemzőknél a függőségi elemzőkéhez képest.

A sebességnek azonban ára van. A konstituenselemzők pontosabbak, mint a függőségi elemzők. Erre épül például az „uptraining" eljárás [10] is, ami a lassú konstituenselemzők kimenetéből a gyors, de gyengébb függőségi elemzőnek tanítópéldákat generál. Többen megmutatták (például [10]), hogy ha a konstituenselemző kimenetét átkonvertáljuk függőségi fákká, jobb eredményt kapunk, mint a legjobb függőségi elemzők. Nyitott kérdés azonban, hogy mi ennek az oka:

- Empirikus eredmények csak angolra és kínaira lettek publikálva. A konstituenselemzők fölénye csak a konfigurációs nyelvek jellegzetessége?
- Angolra a függőségi elemzések a konstituensfák automatikus konverziójából születnek. A konverzió zajos vagy információvesztéssel jár?
- A konstituensreprezentáció algoritmikusan jobban tanulható?
- A függőségi elemzők még csak a tinédzser éveiket élik és néhány év múlva pontosságban is utolérik a konstituenselemzőket?

Az utolsó ponthoz kapcsolódóan megjegyezzük, hogy az összehasonlításhoz használt konstituenselemzők[1] 6-7 évvel ezelőttiek, azóta a konstituenselemzők is rengeteget fejlődtek (habár ezek a fejlesztések nem érhetőek el szabadon letölthető kód formájában). Például a [11] munkában mi is bemutattunk egy újszerű módszert, az erdőalapú rangsoroló elemzőnket, amely angol és német nyelvre is 5% hibacsökkenést eredményezett az eddigi legjobb elemzőkhöz képest.

A konstituenselemzés és függőségi elemzés radikálisan különböző módszerek alkalmazását követeli meg. Kidolgoztunk egy hibrid elemzőt is, amely a két megközelítés különbségeit aknázza ki [12]. A módszer jellemzőket nyer ki az konstituenselemzés kimenetéből, amelyeket felhasznál a függőségi elemzés folyamán (és vice versa). Az eljárás meglepően sokat javít a legjobb függőségi elemzőkön,13% hibacsökkenés a függőségi elemzőkhöz és 6% hibacsökkenés a konstituensből konvertált elemzésekhez képest.

5 Mikor jó egy elemző?

Napjainkig a statisztikai elemzőket szinte kizárólag egy – a tanító adatbázishoz lehető legjobban hasonló – kiértékelő adatbázison értékelték/értékelik ki valamilyen metrika alkalmazásával. Ezzel szemben, ha a gyakorlatban szeretnénk szintaktikai elemzést végezni, akkor (i) a célszövegek valószínűleg számos jellemzőjükben eltérnek a tanító

[1] A Brown parser (2005) és a Berkeley Parser (2006)

adatbáziseitól és (ii) a szintaktikai elemzés célja, hogy valamilyen magasabb szintű feladathoz hasznos bemenetet szolgáltasson, míg az alkalmazott mesterséges kiértékelési metrikák nem képesek a *hasznosság*ot mérni. Valós életbeli alkalmazhatóság szempontjából egy elemző akkor „jó”, ha robosztus (azaz különböző típusú vagy különböző forrásból érkező szövegeken is jól működik) és hasznos bemenetet szolgáltat a végalkalmazásoknak.

Véleményünk szerint a fenti két probléma jelentős figyelmet fog kapni a jövőben. Az (i) problémára a doménadaptációs technikák adhatnak megoldást. Például a [13] munkában bemutattuk webes szövegek elemzésére automatikusan adaptált szintaktikai elemzőinket. A (ii) problémával kapcsolatosan jelenleg is folytatunk kísérleteket. Célunk olyan technikák megtalálása, amivel – a szokásos metrikák helyett – egy célfeladatra – jelen esetben a gépi fordítás átrendezési feladatára – tudjuk optimalizálni a szintaktikai elemzőt. Egy ilyen egyszerű technika a *célzott öntanulás* [14]. Itt egy szintaktikai elemző egy mondathoz tartozó 100 legjobb elemzését kiértékeljük a célfeladathoz való hasznosság szerint (konkrét példánknál az elemzési fák alapján átrendezzük a forrásmondat szavait, majd az átrendezés jóságát számszerűsítjük egy párhuzamos korpusz automatikus szóösszerendelései alapján), majd a leghasznosabb elemzést mint tanítópéldát felhasználva újratanítjuk a szintaktikai elemzőt. Azt kapjuk eredményül, hogy míg a belső metrikák szerint az elemzések rosszabbak, a célfeladat számára azok mégis hasznosabbak.

6 Konklúzió

Jelen munkában tárgyaltuk a statisztikai szintaktikai elemzés fontosabb nyitott kutatási kérdéseit, a magyar szintaktikai elemzők fejlesztéséhez hasznos szempontok szem előtt tartásával.

Bemutattuk a tipológia ún. konfigurációs tengelyét, amelynek egyik végén az erősen konfigurációs angol, míg másik végen a szabad szórendű magyar található. Érveltünk amellett, hogy a magyar nyelvre kidolgozott elemzési módszerek, a gazdag morfológia miatt más – a konfigurációs spektrum közepére elhelyezhető – nyelvek elemzéséhez is hasznos tanulságokkal szolgálhatnak.

Bemutattuk azt, hogy a magyar nyelv statisztikai szintaktikai elemzése nem mondható nehezebbnek, mint bármely más nyelvé, de az elemzők továbbfejlesztéshez nyelvspecifikus, illetve annotációs irányelvekre specifikus problémák megoldása szükséges. Mivel a Szeged Dependencia Treebank statisztikai elemzése kapcsán azt láttuk, hogy a virtuális csomópontok kezelése egy igen gyakori hibaforrás, ezért kidolgoztunk három különböző módszert a virtuális csomópontok automatikus beszúrására.

Tárgyaltuk továbbá a két legelterjedtebb szintaktikai reprezentációt, a konstituens- és függőségi reprezentáció előnyeit és hátrányait. Nem törtünk lándzsát egyik megközelítés mellett sem, célunk az volt, hogy rávilágítsunk: ezidáig senki sem bizonyította, hogy egyik módszer előnyösebb lenne, mint a másik. Bemutattuk továbbá hibrid szintaktikai elemzőnket, amely a két módszer különbözőségeit aknázza ki.

Végül röviden érveltünk a belső kiértékelési metrikák ellen, hiszen a valós életben a tanító adatbázis szövegeitől eltérő szövegeket kell elemeznünk és a végcélunk nem egy jó elemzés elkészítése, hanem olyan elemzések produkálása, amelyek hasznos bemenetül szolgálnak egy célalkalmazás számára.

Köszönetnyilvánítás

A kutatás a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt keretében az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett valósult meg.

Hivatkozások

1. Tsarfaty, R., Seddah, D., Kübler, S., Nivre, J.: Parsing Morphologically Rich Languages: Introduction to the Special Issue. Computational Linguistics (megjelenés előtt)
2. Fraser, A., Schmid, H., Farkas, R., Wang, R., Schütze, H.: Knowledge Sources for Constituent Parsing of German, a Morphologically Rich and Less-Configurational Language. Computational Linguistics (megjelenés előtt)
3. Tsarfaty, R., Seddah, D., Goldberg, Y., Kübler, S., Candito, M., Foster, J., Versley, Y., Rehbein, I., Tounsi, L.: Statistical parsing of morphologically rich languages (spmrl): What, how and whither. In: Proceedings of the NAACL Workshop on Statistical Parsing of Morphologically Rich Languages (2010)
4. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 Shared Task on Dependency Parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007 (2007)
5. Farkas, R., Vincze, V., Schmid, H.: Dependency Parsing of Hungarian: Baseline Results and Challenges. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2012) (2012)
6. Trón, V., Halácsy, P., Rebrus, P., Rung, A,, Vajda, P., Simon, E.: Morphdb.hu: Hungarian lexical database and morphological grammar. In: Proceedings of 5th International Conference on Language Resources and Evaluation (2006)
7. Vincze, V., Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (2010)
8. Seeker, W., Farkas, R., Bohnet, B., Schmid, H., Kuhn, J.: Data-driven Dependency Parsing With Empty Heads. In: Proceedings of the 24th International Conference on Computational Linguistics (2012)
9. Rambow, O.: The Simple Truth about Dependency and Phrase Structure Representations: An Opinion Piece. In: Proceedings of HLT-NAACL (2010)
10. Petrov, S., Chang, P.-C., Ringgaard, M., Alshawi, H.: Uptraining for Accurate Deterministic Question Parsing. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP) (2010)
11. Farkas, R., Schmid, H.: Forest Reranking through Subtree Ranking. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2012) (2012)

12. Farkas, R., Bohnet, B.: Stacking of Dependency and Phrase Structure Parsers. In: Proceedings of the 24th International Conference on Computational Linguistics (2012)
13. Bohnet, B., Farkas, R., Çetinoğlu, Ö.: SANCL 2012 Shared Task: The IMS System. In: Description Notes of the 2012 Shared Task on Parsing the Web (2012)
14. Katz-Brown, J., Petrov, S., McDonald, R., Och, F., Talbot, D., Ichikawa, H., Seno, M., Kazawa, H.: Training a Parser for Machine Translation Reordering. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP) (2011)

VI. Szemantika

A lehetőséghalmazok meghatározása az inkvizitív szemantikában

Szécsényi Tibor

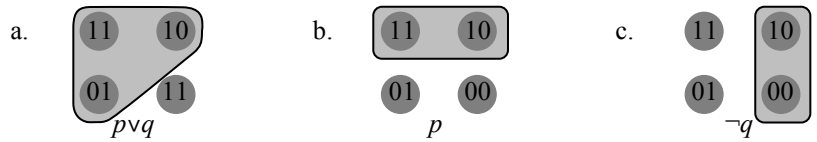
Szegedi Tudományegyetem
Általános Nyelvészeti Tanszék

szecsenyi@hung.u-szeged.hu

Kivonat: Az inkvizitív szemantikában a mondatok interpretációja egy lehetőséghalmaz, amely lehetőségek a lehetséges világok indexeinek egy-egy halmazai. A tanulmány célja az, hogy javaslatot tegyen egy tetszőleges kijelentéslogikai kifejezéshez tartozó lehetőséghalmaz meghatározására a kijelentést alkotó részkijelentések lehetőségeinek terminusában. Az így kapott módszerrel lehetővé válik az olyan diskurzusoknak a dinamikus szemantikai modellezése is, amelyek nem csak információközlő állításokat tartalmaznak, hanem eldöntendő kérdéseket is.
Kulcsszavak: szemantika, inkvizitív szemantika, logika, kijelentéslogika, lehetőségek, lehetséges világok, diskurzus, eldöntendő kérdés

1 A lehetséges világoktól az inkvizitív szemantikáig

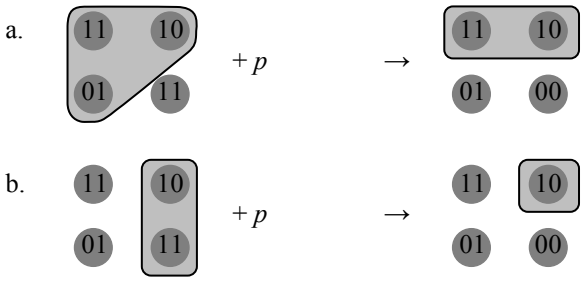
A mondatok jelentését hagyományosan azok információs tartalmával lehet azonosítani. Az *Esik az eső vagy fúj a szél* mondat jelentését az őt alkotó *esik az eső* (=*p*) és *fúj a szél* (=*q*) elemi kijelentések jelentésének ismeretében adhatjuk meg, nevezetesen hogy azokban az esetekben/világokban, amelyekben a *p* és *q* igaz vagy hamis, igaz-e a kérdéses mondat. Két elemi kijelentés esetében négy ilyen indexikus lehetséges világ adódik: lehet olyan világ, amelyikben *p* is és *q* is igaz (jelöljük az ilyen világokat 11 indexszel), vagy lehet csak a *p* igaz, de *q* hamis (10), vagy pedig *q* igaz, de *p* hamis (01), esetleg mindkettő hamis (00). A *p*∨*q* állítás (≈ *esik az eső vagy fúj a szél*) ezek közül háromban igaz (1a), a *p* (≈ , *esik az* eső), illetve a ¬*q* (≈ *nem fúj a szél*) állítások pedig kettőben-kettőben (1b, illetve 1c).



1. ábra. A *p*∨*q*, a *p* és a ¬*q* állítások információs tartalmának a reprezentációi.

A mondatok, megnyilatkozások azonban nem csak önmagukban állnak, hanem egymást követik, leírásokat, diskurzusokat alkotnak. Ekkor már nem (csak) a mondatok különálló információs tartalmát vizsgálhatjuk, hanem a diskurzus egészének az

információs tartalmát. A dinamikus szemantika [4] a mondatok jelentését nem önmagában vizsgálja, hanem az információs tartalom megváltoztatásának a módjaként. Ha például az (1a) ábrán látható információs állapotban hangzik el a *p* állítás, akkor az új információs állapot a (2a) ábrán látható lesz, míg ha az (1c) a jelenlegi információs állapot, akkor ugyanez az állítás a (2b) állapotot eredményezi:



2. ábra. A *p* állítás információsállapot-megváltoztató képessége különböző kiinduló információs állapotok esetén.

Ahhoz, hogy meg tudjuk határozni egy mondat információsállapot-megváltoztató képességét, természetesen ismerni kell az állítás saját, statikus információs tartalmát is. Ha a mondat új információt közöl, akkor az új információs állapot a kiinduló állapot és az új információs tartalom metszeteként kapjuk meg az új információs tartalmat, a fenti példákban az (1b) és az (1a), illetve (1c) metszeteként kapjuk a 2. ábrán látható információs állapotokat.

Az inkvizitív szemantika (Inquisitive Semantics) [2, 3] az információs tartalmon túl azt is igyekszik kézzelfoghatóvá tenni, hogy melyek azok az alapvető lehetőségek, amelyek igazzá tehetik a kijelentést. Az előző példánál maradva, a $p \vee q$ állítás igazzá tételhez két lehetőség adódik, akár a *p* állítás igazsága esetén (11 és 10 indexek), akár a *q* állítás igazsága esetén (11 és 01 indexek) igaz lesz a $p \vee q$ állítás (3a ábra).

3. ábra. A $p \vee q$ és a $p \vee \neg p$ állításokat igazzá tevő lehetőségek az inkvizitív szemantikában.

Mint látható, a diszjunktív állítások inkvizitív tartalma a diszjunkcióban részt vevő állítások mindegyikét egy-egy lehetőségként értelmezi, és a lehetőségek uniója megadja a komplex állítás információs tartalmát. Az inkvizitív tartalom ilyen meghatározása azokban az esetekben is működik, amikor a lehetőségek kizárják egymást, azaz a különböző lehetőségek különböző indexeket tartalmaznak. Erre láthatunk példát a (3b) ábrán, ahol a $p \vee \neg p$ állítás igazzá tevő lehetőségek figyelhetők meg. Az ábrán látható két lehetőség uniója, vagyis a $p \vee \neg p$ kijelentés információs tartalma az összes lehetséges világra kiterjed, azaz a kijelentés elhangzása nem változtatja a diskurzus információs állapotát, ugyanakkor egyértelműen két csoportra osztja a lehetséges világokat, választást kínál fel a két lehetőség közül. Így a diskurzus következő lépésében a két felkínált lehetőség közül lehet választani, azaz a (3b) lehetőséghalmaz a

nyelv eldöntendő kérdéseinek az inkvizitív szemantikai megfelelői: az *esik-e az eső?* vagy az *igaz-e p?* megfelelői, jelölése az inkvizitív szemantikában: ?*p*.

Az inkvizitív szemantika tehát lehetőséget nyújt ahhoz, hogy a diskurzusokat egységes formális eszközökkel tudjuk kezelni, függetlenül attól, hogy a diskurzusban állítást közlő vagy kérdő megnyilatkozások találhatóak-e.

2 Az inkvizitív szemantika formális definíciója

Az inkvizitív szemantika alapfogalmai az *index*, *állapot* (state) és a *lehetőség* (possibility). Az indexek a lehetséges világoknak felelnek meg, az összes indexet tartalmazó halmaz jelölése: *I*. Az állapotok az indexek egy halmaza, ha $\alpha \subseteq I$, akkor α egy állapot. A kijelentések (mondatok) és az állapotok összekapcsolására az *alátámasztás* (support, $\vDash$) fogalmát használjuk: egy állapot alátámaszthat egy kijelentést. Groenendijk és Roelofsen ([3]) definíciója a következő (σ, τ - állapot; *v* - index; *p*, φ, ψ - kijelentés):

(1)

1. $\sigma \vDash p$ *iff* $\forall v \in \sigma : v(p) = 1$
2. $\sigma \vDash \neg\varphi$ *iff* $\forall \tau \subseteq \sigma : \tau \nvDash \varphi$
3. $\sigma \vDash \varphi \vee \psi$ *iff* $\sigma \vDash \varphi$ vagy $\sigma \vDash \psi$
4. $\sigma \vDash \varphi \wedge \psi$ *iff* $\sigma \vDash \varphi$ és $\sigma \vDash \psi$
5. $\sigma \vDash \varphi \rightarrow \psi$ *iff* $\forall \tau \subseteq \sigma$: ha $\tau \vDash \varphi$ akkor $\tau \vDash \psi$

A lehetőségeket az állapotok segítségével határozzuk meg: α egy φ kijelentéshez tartozó *lehetőség*, ha α egy maximális, φ-t alátámasztó állapot (azaz nem valódi részhalmaza egyetlen φ-t alátámasztó állapotnak sem). Egy φ kijelentés inkvizitív szemantikabeli jelentése megegyezik a φ-hez tartozó lehetőségek halmazával.

Érdemes kiemelni, hogy az inkvizitív szemantikában nem feltétlenül érvényesek a hagyományos kijelentéslogika azonosságai. A 2. szabály szerint egy φ kijelentés tagadását az a σ állapot támasztja alá, amelyik maximális a φ-t alá nem támasztó állapotok közül. Így φ-hez hiába is tartozik több lehetőség, a $\neg\varphi$-hez tartozó lehetőség egyetlen lehetőséghez fog járulni, azaz $\neg\varphi$-nek nem lesz inqvizitív tartalma, csak informatív tartalma. A $\neg(p \wedge q)$-hez tehát csak egyetlen lehetőség fog tartozni (4a ábra), míg a klasszikus DeMorgan-azonosságbeli párjához, $\neg p \vee \neg q$-hoz kettő (4b ábra). Tetszőleges több lehetőséget is megengedő φ esetében pedig $\neg\neg\varphi$ szintén egy lehetőséget enged meg, mégpedig pontosan φ lehetőségeinek az unióját.

4. ábra. A $\neg(p \wedge q)$, a $\neg p \vee \neg p$ állításokat igazzá tevő lehetőségek az inkvizitív szemantikában.

Az alátámasztás és a lehetőség fogalmának ez az indirekt definíciója azonban nem nyújt explicit utasítást arra, hogy hogyan lehet egy összetett kijelentés feltételeit hatékonyan meghatározni. Egy $\alpha = \varphi \vee \psi$ összetett kifejezés esetében például minden $\sigma \subseteq \omega$ állapot esetében külön meg kell állapítani, hogy azok alátámasztják-e φ-t, illetve ψ-t

(ha ezek szintén összetett kifejezések, akkor ezt rekurzívan kell ismételni), majd ezek alapján lehet azonosítani a megfelelőek közül a maximálisakat. Ennek a számítási igénye a figyelembe veendő atomi kijelentések számával duplán exponenciálisan arányos. Egy működő implementáció esetében a lehetőségek meghatározásához egy ennél hatékonyabb módszerre van szükség.

3 A lehetőséghalmaz meghatározása relációként értelmezett állapotokkal

Balogh Kata a lehetőséghalmazok meghatározásának egy másik módját adja meg [1].

Balogh az állapotokat az I indexhalmaz valamely részhalmazán értelmezett reflexív és szimmetrikus relációként értelmezi (17. o.):

> Az s állapot egy, az I indexhalmazon értelmezett reflexív és szimmetrikus reláció. (2)

Ekkor egy adott φ kijelentéshez tartozó állapotot ($s[\varphi]$) a következő definíció alapján lehet meghatározni (i és j indexek, p elemi kijelentés, φ és ψ kijelentések, $i(p)$ a p kijelentés igazságértéke az i indexű világban) (19. o.):

$$
\begin{aligned}
&1.\ s[p] = \{\langle i; j\rangle \mid i(p) = 1 \wedge j(p) = 1\} \\
&2.\ s[\neg\varphi] = \{\langle i; j\rangle \mid \langle i; i\rangle \notin s[\varphi] \wedge \langle j; j\rangle \notin s[\varphi]\} \\
&3.\ s[\varphi \vee \psi] = s[\varphi] \cup s[\psi] \\
&4.\ s[\varphi \wedge \psi] = s[\varphi] \cap s[\psi] \\
&5.\ s[\varphi \rightarrow \psi] = \{\langle i; j\rangle \mid \forall \pi \in \{i, j\}^2 : \pi \in s[\varphi] \Rightarrow \pi \in s[\psi]\}
\end{aligned}
\tag{3}
$$

A φ kijelentéshez tartozó állapotot a definíció alapján az őt alkotó részkijelentések állapotaiból közvetlenül meg lehet határozni, csak az azt alkotó rendezett indexpárokat kell figyelembe venni.

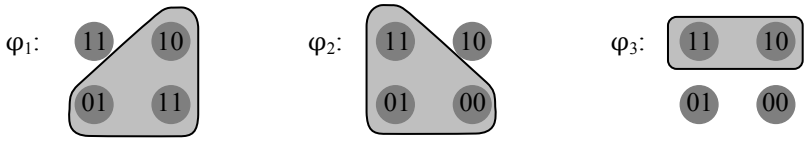
A kijelentéseket igazzá tevő lehetőségeket az így kapott állapotokból vezeti le Balogh (19. o.):

> ρ lehetőség s-ben, (akkor és csakis akkor) ha (4)
>
> 1. $\rho \subseteq I$
> 2. $\forall i,j \in I : \langle i; j\rangle \in s$
> 3. $\neg\exists \rho' : \rho'$ teljesíti az 1. és a 2. feltételt, és $\rho \subset \rho'$

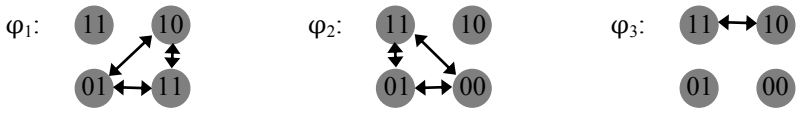
azaz ρ lehetőség s-ben, ha ρ egy maximális, összefüggő részhalmaza s-nek.

Ezekkel a definíciókkal nagyobb hatékonysággal lehet meghatározni az egy kijelentéshez tartozó lehetőségeket, ugyanakkor könnyű belátni, hogy az így definiált lehetőségfogalom nem azonos az inkvizitív szemantika lehetőségfogalmával.

A $\varphi_1 = \neg (p \wedge q)$, $\varphi_2 = \neg (\neg p \wedge q)$ és a $\varphi_3 = q$ kijelentésekhez az 5. ábrán látható, nem több lehetőséget megengedő jelentésreprezentációk tartoznak az inkvizitív szemantika (1) definíciói alapján. Ezek a lehetőségek a (2) és a (4) definíciók alapján csak a 6. ábrán látható relációkból jöhetnek ki (a reflexív tagokat elhagytuk a könnyebb áttekinthetőség kedvéért).

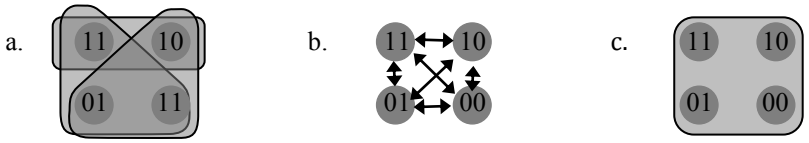


5. ábra. A $\varphi_1 = \neg (p \wedge q)$, $\varphi_2 = \neg (\neg p \wedge q)$ és a $\varphi_3 = q$ kijelentésekhez tartozó lehetőségek az (1) definíció alapján.



6. ábra. A $\varphi_1 = \neg (p \wedge q)$, $\varphi_2 = \neg (\neg p \wedge q)$ és a $\varphi_3 = q$ kijelentésekhez tartozó lehetőségek a (2) és a (4) definíciók alapján.

Az (1.3) definíció alapján azonban a $\varphi_1 \vee \varphi_2 \vee \varphi_3$ kifejezéshez egy három lehetőséget megengedő szemantikai interpretáció tartozik (7a ábra), míg a (3.3) definíciót figyelembe véve ugyanezen kifejezéshez egy olyan *s* állapotreláció tartozik, amelyik megegyezik $I \times I$-vel (7b ábra), ami a (4) definíció szerint csak a mind a négy indexet tartalmazó lehetőséget szolgáltatja (7c ábra).



7. ábra. A $\varphi_1 \vee \varphi_2 \vee \varphi_3$ kijelentéshez tartozó lehetőségek a különböző definíciók alapján.

Látható tehát, hogy a Balogh által az inkvizitív szemantika definiálására ajánlott szabályrendszer [1] nem ugyanolyan logikájú rendszert definiál, mint az eredeti, Groenendijk és Roelofsen által javasolt szabályrendszer [3].

4 A lehetőséghalmaz meghatározása a kifejezésrészek lehetőséghalmazainak figyelembevételével

Ahhoz tehát, hogy az inkvizitív szemantikai reprezentációt számítógépes nyelvészeti alkalmazásokban használhassuk, szükség van arra, hogy egy egyszerű módszert adjunk arra, hogy hogyan lehet egy tetszőleges kijelentés lehetőségeinek a halmazát meghatározni. Erre fogok most egy javaslatot tenni.

A javaslat lényege abban áll, hogy feltételezzük, hogy egy összetett kijelentés alkotórészeinek a lehetőséghalmazai már ismertek. Ezekből a lehetőséghalmazokból létrehozunk halmazelméleti műveletekkel egy lehetőséghalmaz-jelöltet, majd ezen halmazokból kiválasztjuk a maximálisakat. Az összetett kijelentések lehetőséghalmazának a meghatározását a Groenendijk és Roelofsen által javasolt (1) szabályrendszer [3] alapján fogjuk végigtekinteni.

A következő jelöléseket fogom használni:

Ha φ egy kijelentés, akkor
S(φ) a φ-t alátámasztó állapotok halmaza,
P(φ) a φ-hez tartozó lehetőségek halmaza.
Ha α egy tetszőleges állapothalmaz (α ⊆ P(*I*)), akkor
MAX(α) az az állapothalmaz, amely ezek közül a maximálisakat tartalmazza.

4.1 $\sigma \vDash p$ *iff* $\forall \nu \in \sigma : \nu(p) = 1$

Az atomi *p* kijelentéseket azok az állapotok támasztják alá, amelyekben csak olyan indexű lehetséges világok találhatóak, amelyekben az adott *p* kijelentés igaz. Könnyű belátni, hogy ezek közül egyetlenegy maximális található, tehát:

$$P(p) = \{i \in I \mid i(p) = 1\} \tag{5}$$

A további szabályok értelmezésénél azokat az eseteket vizsgáljuk, amikor az összetett kijelentés alkotórészeihez több lehetőség is tartozik, mivel az egyetlen lehetőséggel rendelkező esetek ennek alesetei. Feltételezzük, hogy amennyiben egy kijelentéshez több lehetőség is tartozik, akkor az a kijelentés megadható olyan részkijelentések diszjunkciójaként, amely részkijelentések mindegyikéhez pontosan egy lehetőség tartozik.

4.2 $\sigma \vDash \neg\varphi$ *iff* $\forall \tau \subseteq \sigma : \tau \nvDash \varphi$

σ akkor és csakis akkor lesz a-t alátámasztó állapot, ha σ egyetlen részhalmaza sem támasztja alá φ-t, azaz σ-nak nincs közös eleme a φ-t alátámasztó állapotokkal. Ha a φ kifejezéshez több lehetőség is tartozik, akkor σ ezen lehetőségek (amelyek maguk is állapotok) mindegyikével diszjunkt. Ezen diszjunkt állapotok közül pedig az egyetlen maximálist úgy kapjuk, hogy a φ kifejezéshez tartozó lehetőségek uniójának a komplementerét vesszük:

$$P(\neg\varphi) = I \setminus \bigcup_{\beta \in P(\varphi)} \beta \tag{6}$$

4.3 $\sigma \vDash \varphi \vee \psi$ *iff* $\sigma \vDash \varphi$ vagy $\sigma \vDash \psi$

Két kijelentés diszjunkciója esetében az összetett kifejezést akkor támasztja alá egy állapot, ha a diszjunkcióban szereplő bármely kijelentést is alátámasztja. Ha maguk a részkijelentések is összetett kifejezések, azaz több lehetőség tartozik hozzájuk, akkor ezen lehetőségeknek bármely részhalmaza is alátámasztja a szóban forgó diszjunktív kifejezést. Mivel a φ∨ψ kijelentést alátámasztó maximális állapotok érdekelnek bennünket, ezért nem szükséges a φ-hez és ψ-hez tartozó lehetőségek részhalmazait is megvizsgálni, mivel ezek már nem lesznek maximálisak. Ezek a lehetőségek alá is támasztják az összetett kifejezést, ugyanakkor nem is találhatunk ezeken kívüli alátámasztó állapotokat, tehát elegendő a φ-hez és ψ-hez tartozó lehetőségeket figyelembe venni. Mivel azonban a két kijelentéshez tartozó lehetőségek egymástól függetlenül lettek meghatározva, előfordulhat, hogy az egyik kijelentéshez tartozó lehetőség részhalmaza a másik kijelentéshez tartozó egyik lehetőségnek, ezért még egy

maximalitási vizsgálatot is el kell végezni ahhoz, hogy a φ∨ψ kijelentés lehetőségeit megkapjuk:

$$P(\varphi \vee \psi) = MAX(P(\varphi) \cup P(\psi)) \tag{7}$$

4.4 5.4 σ⊨φ∧ψ *iff* σ⊨φ és σ⊨ψ

A konjunkció első pillantásra egyszerűnek tűnhet, hiszen csak azokat az állapotokat kell megtalálnunk, amelyek a konjunkció részkijelentéseit is alátámasztják. Ha φ-hez és ψ-hez is csak egy-egy lehetőség tartozik, akkor ezek az alátámasztó állapotok a két lehetőség metszetének a részhalmazai lesznek, a maximális pedig maga a metszet. Ha azonban a két részkijelentéshez több lehetőség is tartozik, már elgondolkodtatóbb a helyzet: miknek kell a metszetét/metszeteit venni?

Egyszerűbb esetben, amikor csak az egyik taghoz tartozik több lehetőség, mondjuk φ-hez, ezen lehetőségek bármelyike alátámasztja φ-t, tehát ha a mindkét tagot alátámasztó állapotokat akarjuk előállítani, elegendő a ψ-hez tartozó egyetlen lehetőségnek és a φ-hez tartozó lehetőségeknek a metszeteit előállítani, és ezen metszeteknek a részhalmazait. Ha pedig mindkét taghoz több lehetőség is tartozik a φ-hez tartozó lehetőségeknek kell egyenként a metszetüket venni a ψ-hez tartozó valamennyi lehetőséggel:

$$P(\varphi \wedge \psi) = MAX(\{\alpha \subseteq I \mid \exists\, \alpha_1 \in P(\varphi), \alpha_2 \in P(\psi) : \alpha = \alpha_1 \cap \alpha_2\}) \tag{8}$$

5 Az inkvizitív szemantika további felhasználási lehetőségei

Az inkvizitív szemantikát mindez ideig mint a kijelentéslogika nyelvéhez tartozó szemantikát tekintettük. Azonban könnyű kiterjeszteni elsőrendű predikátumlogikára is, mint ahogy Ivano Ciardelli is tette [2]. Az itt javasolt lehetőséghalmaz-meghatározási módszert is könnyen lehet alkalmazni az elsőrendű logikában, mivel az univerzális, illetve az egzisztenciális kvantor tekinthető a konjunkció, illetve a diszjunkció általánosításának.

Az inkvizitív szemantika kijelentéslogikai alkalmazása melletti egyik legfőbb érv az volt, hogy segítségével nem csak az információközlő állításokhoz, hanem az eldöntendő kérdésekhez is releváns interpretációt lehet rendelni. Ugyanez igaz az elsőrendű változatra is, csak abban az esetben már nem csak az eldöntendő kérdéseket tudjuk egységes keretben kezelni, hanem a kiegészítendő kérdéseket is – ennek pontos kidolgozása és formalizálása azonban még további kutatásokat igényel. Az azonban már most is világos, hogy a kiegészítendő kérdések esetében a mondathoz tartozó lehetőségek több, egymással diszjunkt csoportot alkotnak.

A kiegészítendő kérdések pontos leírásának egyik hozadéka az lehet, hogy segítségükkel a fókuszos mondatok interpretációja is adódik – legalábbis a kimerítő felsorolásos fókuszé. A kiegészítendő kérdésre ugyanis ilyen fókuszos mondatokkal lehet válaszolni, és mint ahogy az eldöntendő kérdés esetében a válasz a két lehetőség közül az egyik alternatívát választja ki, úgy a kiegészítendő kérdésnél is a válaszul elhangzó fókuszos mondat a kérdéshez tartozó lehetőségek egyikét választja ki.

Az inkvizitív szemantikai reprezentáció további alkalmazási területe lehet még a többértelmű kifejezések jelentésének a megadása. Ekkor ugyanis nem kell külön foglalkozni azzal, hogy egy mondatnak vagy nyelvi kifejezésnek több interpretációja is van, hanem egyszerűen vesszük a különböző interpretációkhoz tartozó lehetőségeket, és mint a diszjunkciót tartalmazó kifejezéseknél, a két lehetőséghalmaz unióját képezzük.

Ugyanakkor nem lehet elhallgatni, hogy az inkvizitív szemantikának, csakúgy, mint a lehetséges világok halmazával operáló dinamikus szemantikáknak általában, nagy hátránya, hogy már nem is túlságosan komplex esetekben is hihetetlenül megnő a lehetséges világok száma, így nagyon hamar kezelhetetlenné válik.

Hivatkozások

1. Balogh, K.: Theme with Variations. Doktori értekezés, University of Amsterdam (2009) Letöltve 2012. október 6.: http://www.illc.uva.nl/Publications/Dissertations/DS-2009-07.text.pdf
2. Ciardelli, I.: A first-order inquisitive semantics. In: Aloni, M., Bastiaanse, H., de Jager, T., Schulz, K. (eds): Logic, Language, and Meaning: Selected Papers from the 17th Amsterdam Colloquium. Springer-Verlag, Berlin Heidelberg (2010) 234–243
3. Groenendijk, J., Roelofsen, F.: Inquisitive Semantics and Pragmatics. In: Larrazabal, J.M., Zubeldia, L. (eds): Meaning, Content and Argument, Proceedings of the ILCLI International Workshop on Semantics, Pragmatics and Rhetoric. University of the Basque Country Publication Service (2009) 41–72
4. Groenendijk, J., Stokhof, M.: Dynamic predicate logic. Linguistics and Philosophy, Vol. 14 (1991) 39–100

Magyar és angol szavak szemantikai hasonlóságának automatikus kiszámítása

Dobó András, Csirik János

Szegedi Tudományegyetem, Informatikai Tanszékcsoport,
H-6720 Szeged, Árpád tér 2.
{dobo,csirik}@inf.u-szeged.hu

Kivonat: Szavak szemantikai hasonlóságának vizsgálata számos számítógépes nyelvészeti probléma megoldásában fontos szerepet tölt be. Habár már sok különféle módszer létezik e feladatra, az eredményeken még mindig lehetne javítani. Egy korábbi kutatásunk során olyan módszereket fejlesztettünk ki angol szavak szemantikai hasonlóságának automatikus megállapítására, amelyek nagyméretű statikus korpuszokból kinyert statisztikai információ alapján készítenek bináris vagy numerikus tulajdonságvektorokat a szavakhoz, majd a szavak hasonlóságát vektoraik hasonlóságaként számolják ki. Jelen cikkünkben korábbi módszereink továbbfejlesztett változatát mutatjuk be, melyek a korábbiakhoz képest új vektorhasonlóságokat is felhasználnak, továbbá már alkalmasak magyar szavak közötti szemantikai hasonlóság megállapítására is, mely legjobb tudásunk szerint egyedülálló. Az algoritmusok angol és magyar nyelvű tesztadatbázisokon kiértékelve is versenyképes eredményeket érnek el.

1 Bevezetés

Számos számítógépes nyelvészeti probléma megoldásában, mint például az információkinyerésben, helyesírás-javításban és szójelentés-egyértelműsítésben, szavak szemantikai hasonlóságának az ismerete nagy segítséget nyújthat. Ezért az elmúlt nagyjából 20 évben számos kutatás irányult szavak jelentésbeli hasonlóságának automatikus meghatározására. A legtöbb erre a feladatra kialakított módszer webes kereséseket (pl. Google vagy Yahoo!), illetve lexikai adatbázisokat (pl. WordNet vagy Roget's Thesaurus) alkalmaz a hasonlóság kiszámítására. Ugyan ezek használata sok szempontból előnyös és az őket használó algoritmusok általában jól működnek, mint ahogy azt korábban is bemutattuk [1], sok hátránnyal is rendelkeznek.

Ezért korábbi kutatásunk [1] során olyan módszereket készítettünk, melyek sem webes kereséseket sem lexikai adatbázisokat nem használnak, és pusztán statikus korpuszok felhasználásával képesek angol szavak szemantikai hasonlóságának az automatikus kiszámítására[1]. Ezek a módszerek először létrehoznak egy tulajdonságvektort minden szóhoz a felhasznált korpuszban található környezeti szavak vagy

[1] Habár felhasználtuk a WordNetet szavak lemmájának meghatározására, semmi másra nem használtuk. Ez pedig helyettesíthető lenne egyéb módszerekkel.

nyelvtani kapcsolatok és valamely súlyozási módszer segítségével. Ezután szavak hasonlóságát a vektoraik hasonlóságaként számítják ki.

Jelen cikkünkben e korábbi módszerek továbbfejlesztett változatát mutatjuk be. Ezek a módszerek a már korábban használt egy bináris és kettő numerikus vektorhasonlóság mellett további három numerikus hasonlósági mértéket használnak fel. Továbbá, már nem csak angol, hanem magyar szavak közötti szemantikai hasonlóság megállapítására is alkalmasak, mely legjobb tudásunk szerint egyedülálló. A különálló módszerek mellett azok kombinációit is kipróbáltuk, és a korábbi angol nyelvű tesztadatbázisok mellett magyar nyelvű tesztadatbázisokon is kiértékeltük őket.

A következő szakasz a témához kapcsolódó egyéb kutatásokat foglalja röviden össze. Ez után algoritmusaink bemutatása következik, amit az algoritmus eredményeinek prezentálása és a konklúziók levonása követnek.

2 Kapcsolódó munkák

Habár már számos kutatás vizsgálta angol szavak szemantikai hasonlóságának automatikus megállapítását, legjobb tudásunk szerint a miénk az első olyan módszer, mely magyar szavak szemantikai hasonlóságával foglalkozik. Ezért ebben az alfejezetben az eddig publikált, angol szavak szemantikai hasonlóságának kiszámításával foglalkozó módszereket jellemezzük röviden (részletesebb áttekintésük korábbi cikkünkben található meg [1]). Ezeket a felhasznált adatforrások és a működésük alapján három nagy kategóriába sorolhatjuk.

Sok módszer nagyméretű lexikai adatbázisokban tárolt információt használ fel, és a kinyert információk alapján számolja ki szavak szemantikai hasonlóságát. A legtöbb a WordNetet használja, de léteznek olyanok is, melyek a Roget's Thesaurust. Egy nagyon jó példa erre Tsatsaronis et al. [2] módszere, mely egy WordNet alapú hasonlósági pontszámot definiál. Ennek a kiszámításához figyelembe veszi a szavak WordNetbeli távolságát, a közöttük lévő szavak WordNetbeli mélységét és a szavak közti kapcsolatok típusait. Módszerüket kibővítették, hogy ne csak szavak, hanem hosszabb szövegrészek hasonlóságának megállapítására is alkalmas legyen.

Más módszerek szavak hasonlóságának becsléséhez webes kereséseket indítanak a vizsgált szavakkal, és a visszaadott találatok számát, valamint a visszaadott szövegtöredékeket használják fel. Például Higgins [3] webes kereséseket indít a vizsgált szavakkal külön-külön és együtt is, majd a hasonlóságukat a visszaadott találatok számából kiszámított pontonkénti kölcsönös információként adja meg.

Léteznek olyan módszerek is, melyek egy tulajdonságvektort képeznek minden szóhoz a szó egy nagyméretű korpuszban talált környezetei alapján. Habár a mi módszereink hasonlóak ezekhez a módszerekhez, a mieink új tulajdonságokat, súlyozási módszereket és vektorhasonlósági mértékeket használnak a már korábban is alkalmazottak mellett. Egy ebbe a kategóriába tartozó módszer például Rappé [4] is, mely minden szóhoz egy numerikus tulajdonságvektort készít a szó megtalált előfordulási környezetei alapján. Ezekben a vektorokban azok a környezeti szavak találhatók meg, melyek a vizsgált szótól legfeljebb két szó távolságra találhatóak a korpuszban, és a súlyuk olyan jól ismert szókapcsolati mértékeken alapszik, mint a pontonkénti kölcsönös információ. A vektorok által adott mátrixot ezután összetömöríti az SVD mód-

szer segítségével. Végül a szavak hasonlósága a tömörített vektoraik hasonlóságaként kerül kiszámításra.

Mindhárom fő módszertípusnak megvannak az előnyei és a hátrányai, ezért sok kutatás oly módon próbálta meg az addig elért eredményeket tovább javítani, hogy különböző típusú módszereket kombinált, így próbálva azok előnyeit ötvözni. Turney et al. [5] módszere például négy különböző módszer ötvözete. Az első az LSA [6], a második egy webes kereséseken alapuló módszer (PMI-IR), a harmadik egy online fogalomtárban keres (Wordsmyth thesaurus online) és az utolsó webes keresések által visszaadott szövegtöredékeket dolgoz fel. Ezt a négy módszert többféleképpen kombinálták (például a szorzat szabállyal) a végső hasonlóság kiszámításához.

3 Módszereink

Módszereink alapötlete, mint sok egyéb módszer alapötlete, az, hogy a szemantikailag hasonló szavak hasonlóan viselkednek és hasonló szövegkörnyezetekben fordulnak elő. Ezért módszereink minden szóhoz egy tulajdonságvektort képeznek statikus korpuszokból kinyert statisztikai információ alapján. Ezen vektorokban különféle tulajdonságokat, így például a szavak környezetében előforduló úgynevezett környezeti szavakat és a szavakhoz kapcsolódó nyelvtani kapcsolatokat alkalmaznak. Azért, hogy a vektorokon belül a tulajdonságok fontosságát reprezentálni tudják, különféle súlyozásokat alkalmaznak. A szavak hasonlóságát az algoritmusok a létrejött súlyozott vektorok hasonlóságaként definiálják.

A következő alfejezetben korábbi, kizárólag angol szavak szemantikai hasonlóságának számolására alkalmas módszereinket mutatjuk be nagy vonalakban. Ezek a módszerek teljes részletességben már korábbi cikkünkben [1] is bemutatásra kerültek angol nyelven. Ezután rátérünk arra, hogy módszereinket azóta milyen módon fejlesztettük tovább, bővítettük ki.

3.1 Angol szavak szemantikai hasonlóságának kiszámítása

A szavakhoz képzett vektorokban szereplő tulajdonságok kinyerésére két fő változatot használtunk. Az első a szózsák (bag-of-words) alapú megközelítés. Ez a vizsgált szó összes előfordulási helyét megkeresi a felhasznált korpuszban, és az előfordulások környezetében lévő minden, legfeljebb három távolságra szereplő szót belerakja a tulajdonságvektorba, egy távolságalapú súlyozást felhasználva. A másik módja a tulajdonságok kinyerésének a nyelvtani kapcsolatok felhasználása. Ehhez először a korpuszt automatikusan elemeztük a C&C CCG parser [7] segítségével, majd tulajdonságként a vizsgált szóhoz nyelvtanilag közvetlenül kapcsolódó szavakat használtuk a nyelvtani kapcsolatok típusával együtt. Mindkét módszerhez három korpuszt, a British National Corpust (BNC), a Web 1T 5-gram Corpust (csak a 4- és 5-gramokat) és az angol Wikipedia korpuszát használtuk (a Wikipedia korpuszt előfeldolgoztuk Rafael Mudge wikipedia2text_rsm_mods toolkitjével[2]). Mivel tetszőleges korpusz alkalmazható a tulajdonságok kinyeréséhez, ezért a módszereink könnyen adaptálhatók más tárgykörökre és más nyelvekre.

[2] http://blog.afterthedeadline.com/2009/12/04/generating-a-plain-text-corpus-from-wikipedia/

1. táblázat: Módszereink eredménye az angol Miller-Charles adathalmazon (Spearman korreláció). Jelölések: bnc/enwiki/web1t5gram jelöli a korpuszt; bagofwords/parsed jelöli a tulajdonságtípusokat (szózsák vagy nyelvtani kapcsolatok); lin/num jelöli a tulajdonságvektorok létrehozásának és összehasonlításának módszert (Lin [8] módszerén alapuló vagy numerikus vektorokat alkalmazó); cos/dice/pears/spear/zkl jelöli a hasonlósági mértéket; freq/logfreq/ pmi/loglh/qw/pw/rapp jelöli a súlyozást; + jelöli két módszer kombinációját.

Módszer	**Eredmény**
enwiki-parsed-num-zkl-loglh+bnc-bagofwords-num-zkl-loglh	0,773
bnc-bagofwords-num-cos-qw+enwiki-parsed-num-cos-freq	0,773
enwiki-parsed-num-zkl-loglh+enwiki-parsed-num-pears-logfreq	0,754
bnc-bagofwords-num-cos-qw+enwiki-parsed-num-cos-qw	0,750
bnc-bagofwords-num-zkl-loglh	0,744
bnc-parsed-num-pears-qw+enwiki-bagofwords-num-cos-pmi	0,737
bnc-bagofwords-num-zkl-loglh+enwiki-parsed-num-pears-pmi	0,736
bnc-bagofwords-num-zkl-loglh+enwiki-parsed-num-cos-pmi	0,736
enwiki-parsed-num-pears-pmi+enwiki-bagofwords-num-pears-pmi	0,729
enwiki-parsed-num-pears-pmi	0,727
enwiki-parsed-num-cos-pmi	0,727
enwiki-parsed-num-zkl-loglh+enwiki-bagofwords-num-pears-pmi	0,721
enwiki-parsed-num-zkl-loglh+enwiki-bagofwords-num-cos-pmi	0,721
enwiki-parsed-num-zkl-loglh	0,718
bnc-parsed-num-pears-loglh+enwiki-parsed-num-pears-pmi	0,712
bnc-parsed-num-cos-loglh+enwiki-parsed-num-cos-pmi	0,712
enwiki-bagofwords-num-spear-logfreq+enwiki-parsed-num-cos-pmi	0,703
enwiki-bagofwords-num-pears-pmi	0,684
enwiki-bagofwords-num-zkl-loglh	0,548

A tulajdonságvektorok létrehozására és összehasonlítására szintén két különféle szemléletmódot tekintettünk. Először Lin [8] módszerét (azt, amelyik statikus korpuszokkal dolgozik és nem használja fel a WordNetet) újraimplementáltuk néhány módosítással. Ez a módszer bináris tulajdonságvektorokkal dolgozik, melyeket egy Lin [8] által definiált mértékkel hasonlít össze. A másik szemlélet numerikus tulajdonságvektorokkal dolgozik, ahol minden tulajdonsághoz egy súly is tartozik. A súlyok közt szerepelnek egyszerű gyakoriságalapú (gyakoriság - freq, gyakoriság logaritmusa - logfreq), illetve bonyolultabb információelméleti súlyok (pontonkénti kölcsönös információ - pmi, log-likelihood arány - loglh, qw, pw, Rapp-féle [4] - rapp) is. Ez a modell a súlyozott vektorokat különféle vektorhasonlósági mértékekkel (koszinusz hasonlóság - cos, Lin-féle Dice-együttható [8] - dice) hasonlítja össze.

Mivel sok szó többféle szófajt is felvehet, és a különböző szófajú szavakhoz különböző tulajdonságok a fontosak, ezért szavak összehasonlításakor fontos az, hogy

először a szavak szófaját meghatározzuk. Ez módszerünk esetében a tesztszavaknak az adott korpuszban vett előfordulási gyakoriságának felhasználásával történik [1].

Azért, hogy a különféle módszerek előnyeit egyesíteni tudjuk, a módszereket nem csak külön-külön, hanem egymással kombinálva is teszteltük. Két módszer kombinációjakor a szópárok hasonlósága először a két módszerrel külön kerül meghatározásra, majd a kombinált hasonlóság e két hasonlósági pontszámból kerül kiszámításra [1].

3.2 A továbbfejlesztett módszer

Az előző alfejezetben ismertetett módszereinken két fő változtatást hajtottunk végre. Egyrészt a már meglévő három vektorhasonlósági módszer (lin, cos, dice) mellé további három hasonlósági metrikát implementáltunk. Az első a Pearson-féle korrelációs együttható (pears), mely két numerikus változó közti összefüggés erősségét mutatja meg. A másik a Spearman-féle rangkorrelációs együttható (spear), mely a Pearson-együttható olyan speciális esete, ami a numerikus értékek helyett azok rangjával számol. A harmadik megvalósított metrika a Zero-KL metrika [9] inverze (zkl). A Zero-KL metrika a Kullback-Leibler divergencia olyan módosítása, mely már 0 valószínűséget tartalmazó valószínűségi eloszlásokra is értelmezett. Mivel a Zero-KL annál nagyobb értéket vesz fel, minél kevésbé hasonló két valószínűségi eloszlás, és mivel a többi hasonlósági mértékünk pont fordítva működik, ezért mi az inverzét alkalmaztuk.

Az új hasonlósági mértékek alkalmazása mellett még egy nagyon lényeges részét fejlesztettük tovább az algoritmusainknak. Módszereink eddig pusztán angol szavak közötti szemantikai hasonlóság kiszámítására voltak képesek. A továbbfejlesztett változatok már képesek magyar szavak közötti szemantikus hasonlóság automatikus kiszámítására is, melyre legjobb tudásunk szerint jelenleg egyetlen másik módszer sem képes. Magyar tesztszavak esetén módszereink az összehasonlítást pillanatnyilag csak a szózsák modell alapján végzik, vagyis minden tesztszóhoz megkeresik a felhasznált (magyar nyelvű) korpuszban a szó előfordulási helyeit, és az ott talált környezeti szavakat használják fel tulajdonságként, a nyelvtani kapcsolatok figyelembe vétele nélkül. Korpuszként a magyar Wikipédia korpuszát használtuk fel (szintén előfeldolgoztuk Rafael Mudge wikipedia2text_rsm_mods toolkitjével). A jövőben majd szeretnénk megvalósítani a nyelvtani kapcsolatokat alkalmazó modellt is.

4 Eredmények

Az elkészült módszereket mind angol, mind magyar tesztadatbázisokon kiértékeltük. Angol szavak esetén két gyakran alkalmazott adathalmazt használtunk fel. Az első a 30 szópárból álló Miller-Charles adathalmaz (MC), melynél minden szópárhoz 38 egyetemi hallgató rendelt hasonlósági pontszámot. Mivel a korábbi WordNet-verziók nem tartalmaztak két szót e szavakból, ezért rendszerint csak a maradék 28 szópárt használták fel a kiértékelésben, és mi is így tettünk. A másik adathalmaz a 80 kérdésből álló TOEFL szinonimakérdések halmaza, ahol minden kérdés egy tesztszót és négy lehetséges megoldást tartalmaz, a feladat pedig annak eldöntése, hogy melyik szó a leghasonlóbb a tesztszóhoz. A kiértékelési metrika az MC adathalmaz esetén az átlagos pontszámokkal vett Spearman-korreláció, míg a TOEFL adathalmaz esetén a helyes válaszok százaléka volt.

2. táblázat: Módszereink eredménye az angol TOEFL-kérdéseken (helyes válaszok százaléka).

Módszer	**Eredmény**
bnc-parsed-num-pears-loglh+enwiki-parsed-num-pears-pmi	88,75%
bnc-parsed-num-cos-loglh+enwiki-parsed-num-cos-pmi	88,75%
enwiki-parsed-num-pears-pmi+enwiki-bagofwords-num-pears-pmi	87,50%
enwiki-parsed-num-zkl-loglh+enwiki-bagofwords-num-pears-pmi	87,50%
enwiki-parsed-num-zkl-loglh+enwiki-bagofwords-num-cos-pmi	87,50%
bnc-parsed-num-pears-qw+enwiki-bagofwords-num-cos-pmi	86,25%
bnc-bagofwords-num-zkl-loglh+enwiki-parsed-num-pears-pmi	85,00%
bnc-bagofwords-num-zkl-loglh+enwiki-parsed-num-cos-pmi	85,00%
enwiki-parsed-num-zkl-loglh+enwiki-parsed-num-pears-logfreq	83,75%
enwiki-bagofwords-num-pears-pmi	83,75%
enwiki-parsed-num-pears-pmi	82,50%
enwiki-parsed-num-cos-pmi	82,50%
enwiki-bagofwords-num-spear-logfreq+enwiki-parsed-num-cos-pmi	82,50%
enwiki-bagofwords-num-zkl-loglh	81,25%
enwiki-parsed-num-zkl-loglh	80,00%
enwiki-parsed-num-zkl-loglh+bnc-bagofwords-num-zkl-loglh	80,00%
bnc-bagofwords-num-cos-qw+enwiki-parsed-num-cos-qw	77,50%
bnc-bagofwords-num-zkl-loglh	72,50%
bnc-bagofwords-num-cos-qw+enwiki-parsed-num-cos-freq	72,50%

Mivel magyar szavakra tudomásunk szerint nem létezik még olyan algoritmus, mely szavak szemantikai hasonlóságának megállapítására képes, ezért még nincs általánosan használt tesztadatbázis sem a kiértékeléshez. Ennek hiányában arra a következtetésre jutottunk, hogy legegyszerűbben oly módon tudjuk módszereinket kiértékelni, hogy az angol szavakat tartalmazó két tesztadatbázist lefordítjuk magyarra. Ugyan tudjuk, hogy a legtöbb angol szóhoz nem létezik olyan magyar szó, mely pontosan ugyanazzal a jelentéskörrel rendelkezik, mégis úgy gondoljuk, hogy kezdeti kiértékelésre megfelelőek ezek az adatbázisok, és hogy segítségükkel algoritmusaink teljesítménye jól becsülhető. Így végül magyarra az MC adathalmaz magyar fordítását (MC-Hu), illetve a TOEFL adathalmaz magyar fordítását (TOEFL-Hu) használtuk fel, az angollal megegyező kiértékelési metrikák használatával. A fordításnál igyekeztünk, hogy a magyar tesztek minél jobban tükrözzék angol verzióik tulajdonságait.

Az algoritmusok angol tesztszavakon adott eredményeit az 1. és 2. táblázat foglalják össze. Az algoritmusaink által elért legjobb eredmény az MC adathalmaz esetén 0,773, míg a TOEFL-kérdések esetén 88,75% volt. Ha összehasonlítjuk az új vektorhasonlóságokat alkalmazó módszerek eredményét a régiekével, akkor jól látható, hogy az újabb verziók hasonlóan jó eredményt értek el, mint korábbi társaik, sőt néhol a korábbiaknál jobbat. A legtöbb olyan algoritmus, mely jól teljesített az egyik adat-

halmazon, az jó eredményt ért el a másikon is. Néhányat kiemeltünk azok közül, melyek a két adathalmazt együttesen figyelembe véve a legjobb eredményt érték el:

a. enwiki-parsed-num-zkl-loglh+bnc-bagofwords-num-zkl-loglh:
   (MC: 0,773, TOEFL: 80,00%)
b. enwiki-parsed-num-zkl-loglh+enwiki-parsed-num-pears-logfreq:
   (MC: 0,754, TOEFL: 83,75%)
c. bnc-parsed-num-pears-loglh+enwiki-parsed-num-pears-pmi:
   (MC: 0,712, TOEFL: 88,75%)
d. enwiki-parsed-num-pears-pmi+enwiki-bagofwords-num-pears-pmi:
   (MC: 0,729, TOEFL: 87,50%)
e. bnc-parsed-num-pears-qw+enwiki-bagofwords-num-cos-pmi:
   (MC: 0,737, TOEFL: 86,25%)

3. táblázat: Eredményeink összehasonlítása más módszerek eredményeivel az angol Miller-Charles adathalmazon (Spearman-korreláció).

Módszer	**Eredmény**	**Felhasznált adatforrások**
Emberi felső korlát [11]	0,934	
Agirre et al. [10]	0,92	WordNet, korpusz
Patwardhan és Pedersen [12]	0,91	WordNet
Jarmasz és Szpakowicz [13]	0,87	Roget's Thesaurus
Tsatsaronis et al. [2]	0,856	WordNet
Kulkarni és Caragea [14]	0,835	Webes keresés
Lin [8]	0,82	WordNet, korpusz
Resnik [11]	0,81	WordNet, korpusz
enwiki-parsed-num-zkl-loglh+ bnc-bagofwords-num-zkl-loglh	0,773	korpusz
enwiki-parsed-num-zkl-loglh+ enwiki-parsed-num-pears-logfreq	0,754	korpusz
bnc-parsed-num-pears-qw+ enwiki-bagofwords-num-cos-pmi	0,737	korpusz
bnc-bagofwords-num-zkl-loglh+ enwiki-parsed-num-pears-pmi	0,736	korpusz
enwiki-parsed-num-pears-pmi+ enwiki-bagofwords-num-pears-pmi	0,729	korpusz
enwiki-parsed-num-pears-pmi	0,727	korpusz
Gabrilovich és Markovitch [15]	0,72	korpusz
bnc-parsed-num-pears-loglh+ enwiki-parsed-num-pears-pmi	0,712	korpusz
Milne és Witten [16]	0,70	Wikipedia linkek, Webes keresés
Sahami és Heilman [17]	0,618	Webes keresés

Eredményeinket mások módszereivel a 3. és 4. táblázatban hasonlítottuk össze. Ez azt mutatja, hogy módszereink az MC adathalmazon általában közepes eredményt értek el, míg a TOEFL adathalmazon összességében harmadik legjobban teljesítettek. Azonban, ha csak azokat a módszereket tekintjük, melyek a mi módszereinkhez hasonlóan csak statikus korpuszokat használnak fel adatforrásként, akkor több módszerünk is (például d. és e.) az MC és a TOEFL adathalmazon rendre első és második legjobb eredményt ért el más kutatások eredményeihez hasonlítva.

Az 5. és 6. táblázat foglalja össze algoritmusaink eredményét a magyar tesztadatbázisokon. Az MC-Hu adatbázis esetén elért legjobb eredmény 0,637, míg a TOEFL-Hu kérdések esetén 60,00%. Ebben az esetben azonban korábbi eredmények hiányában nem tudjuk eredményeinket másokéval összehasonlítani. Viszont, ha ezeket az eredményeket az angol tesztadatbázisokon elért eredményekkel vetjük össze, akkor az figyelhető meg, hogy magyar tesztszavakon átlagosan lényegesen rosszabb eredményt értek el, mint az angol tesztek esetén. Véleményünk szerint ez több tényezőnek tudható be. Egyrészt a magyar nyelv nyelvtana lényegesen bonyolultabb az angolénál. Másrészt a felhasznált magyar korpusz mérete lényegesen kisebb az alkalmazott angol korpuszokénál. Harmadrészt, mivel a magyar nyelv szabad szórendű, ezért a nyelvtani kapcsolatok sokkal több információval szolgálnának egy szóról, mint a környezeti szavak. Tehát a nyelvtani kapcsolatokat is felhasználó modell véleményünk szerint az eddigieknél jobb eredményeket érhetne el.

A magyar nyelv esetén is azok az algoritmusok, melyek az egyik adathalmazon jól teljesítettek, általában jó eredményt értek el a másikon is. A következő algoritmusok teljesítettek legjobban mindkettő adatbázist figyelembe véve:

f. huwiki-bagofwords-num-zkl-loglh+huwiki-bagofwords-num-pears-pmi:
(MC: 0,637, TOEFL: 58,75%)
g. huwiki-bagofwords-num-zkl-pmi+huwiki-bagofwords-num-pears-pmi:
(MC: 0,629, TOEFL: 57,50%)
h. huwiki-bagofwords-num-zkl-loglh:
(MC: 0,622, TOEFL: 60,00%)

Megvizsgáltuk azt is, hogy melyek azok a módszerek, melyek a felhasznált korpusztól és a nyelvtől függetlenül jól teljesítenek. Mivel a különböző nyelvekhez más korpuszok tartoznak, ezért a korpuszokat sem vettük figyelembe. Az találtuk, hogy mind kombinált, mind különálló módszerből létezik számos olyan, mely jól teljesít mindkét nyelv mindkét tesztadatbázisa esetén, vagyis nyelvtől és tesztadatbázistól függetlenül jól tud működni. Ezek közül néhány:

i. num-zkl-loglh+num-pears-pmi:
(MC: 0,736, TOEFL: 87,50%, MC-Hu: 0,637, TOEFL-Hu: 58,75%)
j. num-zkl-loglh+num-cos-pmi:
(MC: 0,736, TOEFL: 87,50%, MC-Hu: 0,611, TOEFL-Hu: 58,75%)
k. num-zkl-loglh:
(MC: 0,744, TOEFL: 81,25%, MC-Hu: 0,622, TOEFL-Hu: 60,00%)
l. num-pears-pmi:
(MC: 0,727, TOEFL: 83,75%, MC-Hu: 0,617, TOEFL-Hu: 58,75%)

A felsorolt négy algoritmus mindegyike jól teljesít mind a négy tesztet tekintve. Ha csak azokat az algoritmusokat vesszük figyelembe, amelyek kizárólag statikus korpuszokat használnak fel adatforrásként, akkor az i. és j. algoritmus által elért eredmények például az MC és TOEFL adathalmazon tesztelve rendre az első és második legjobbak más kutatások eredményeihez hasonlítva, továbbá az MC-Hu és TOEFL-Hu adathalmazokon elért eredményeik is saját módszereink eredményeit tekintve a legjobbak között vannak.

4. táblázat: Eredményeink összehasonlítása más módszerek eredményeivel az angol TOEFL kérdéseken (helyes válaszok százaléka).

Módszer	**Eredmény**	**Felhasznált adatforrások**
Turney et al. [5]	97,5%	Webes keresés, fogalomtár
Rapp [4]	92,5%	korpusz
bnc-parsed-num-pears-loglh+ enwiki-parsed-num-pears-pmi	88,75%	korpusz
enwiki-parsed-num-pears-pmi+ enwiki-bagofwords-num-pears-pmi	87,50%	korpusz
enwiki-parsed-num-zkl-loglh+ enwiki-bagofwords-num-pears-pmi	87,50%	korpusz
Tsatsaronis et al. [2]	87,5%	WordNet
bnc-parsed-num-pears-qw+ enwiki-bagofwords-num-cos-pmi	86,25%	korpusz
Matveeva et al. [18]	86,25%	korpusz
enwiki-parsed-num-zkl-loglh+ enwiki-parsed-num-pears-logfreq	83,75%	korpusz
enwiki-parsed-num-pears-pmi	82,50%	korpusz
Higgins [3]	81,3%	Webes keresés
enwiki-parsed-num-zkl-loglh+ bnc-bagofwords-num-zkl-loglh	80,00%	korpusz
Jarmasz és Szpakowicz [13]	78,7%	Roget's Thesaurus
Átlagos nem angol anyanyelvű, amerikai egyetemre felvételiző diák [6]	64,5%	
Landauer és Dumais [6]	64,3%	korpusz
Lin [8]	24,0%	WordNet, korpusz
Resnik [11]	20,3%	WordNet, korpusz

5. táblázat: Módszereink eredménye a magyar Miller-Charles adathalmazon (Spearman-korreláció).

Módszer	**Eredmény**
huwiki-bagofwords-num-zkl-loglh+ huwiki-bagofwords-num-pears-pmi	0,637
huwiki-bagofwords-num-zkl-pmi+ huwiki-bagofwords-num-pears-pmi	0,629
huwiki-bagofwords-num-zkl-loglh	0,622
huwiki-bagofwords-num-zkl-logfreq+ huwiki-bagofwords-num-pears-pmi	0,621
huwiki-bagofwords-num-pears-pmi	0,617
huwiki-bagofwords-num-zkl-loglh+ huwiki-bagofwords-num-cos-pmi	0,611
huwiki-bagofwords-num-cos-pmi	0,610
huwiki-bagofwords-num-pears-pmi+ huwiki-bagofwords-num-cos-freq	0,588

6. táblázat: Módszereink eredménye a magyar TOEFL-kérdéseken (helyes válaszok százaléka).

Módszer	**Eredmény**
huwiki-bagofwords-num-zkl-loglh	60,00%
huwiki-bagofwords-num-pears-pmi+ huwiki-bagofwords-num-cos-freq	60,00%
huwiki-bagofwords-num-pears-pmi	58,75%
huwiki-bagofwords-num-zkl-logfreq+ huwiki-bagofwords-num-pears-pmi	58,75%
huwiki-bagofwords-num-zkl-loglh+ huwiki-bagofwords-num-pears-pmi	58,75%
huwiki-bagofwords-num-zkl-loglh+ huwiki-bagofwords-num-cos-pmi	58,75%
huwiki-bagofwords-num-zkl-pmi+ huwiki-bagofwords-num-pears-pmi	57,50%
huwiki-bagofwords-num-cos-pmi	57,50%

5 Konklúzió

Cikkünkben olyan módszereket mutattunk be, melyek alkalmasak magyar és angol szavak közötti szemantikai hasonlóság automatikus megállapítására. Ezek statikus korpuszokból kinyert statisztikai információk alapján egy tulajdonságvektort képeznek minden szóhoz, majd a szavak hasonlóságát vektoraik hasonlóságaként számolják ki. Több variációt kipróbáltunk, melyek különféle tulajdonságtípusokat, vektortípuso-

kat, súlyozásokat, valamint vektorhasonlósági mértéket alkalmaznak, továbbá a különálló módszerek kombinációit is teszteltük.

Minden módszert nyelvenként két különböző adathalmazon értékeltünk ki, angol esetén a Miller-Charles adathalmazon (MC) és a TOEFL szinonimakérdéseken, magyar esetén pedig ezek magyarra fordított változatán (MC-Hu és TOEFL-Hu). Angol szavak esetén legjobb módszereink közepes eredményt értek el az MC adathalmazon, míg harmadik legjobban teljesítettek a TOEFL-kérdéseken. Azonban, ha kizárólag azokat a módszereket tekintjük, melyek csak statikus korpuszokat alkalmaznak, akkor algoritmusaink a két adathalmazon rendre első és második legjobb eredményt értek el.

Az algoritmusok angol tesztszavakon lényegesen jobb eredményt értek el, mint magyar változataikon. Ezt részben annak tudjuk be, hogy a magyar nyelv nyelvtana lényegesen bonyolultabb az angolénál és hogy a felhasznált magyar korpusz mérete lényegesen kisebb az alkalmazott angol korpuszokénál. Továbbá, mivel a magyar nyelv szabad szórendű, ezért a nyelvtani kapcsolatok sokkal több információval szolgálnának egy szóról, mint az általunk jelenleg használt környezeti szavak. Ezért véleményünk szerint a nyelvtani kapcsolatokat is felhasználó modell az eddigieknél lényegesen jobb eredményeket érhetne el.

Az eredmények alapján úgy gondoljuk, hogy módszereink sikeresen alkalmazhatóak lennének valós problémákon is. Megfigyelhető, hogy az algoritmusok (főként az angol nyelv esetén) jobb eredményt érnek el a TOEFL-kérdéseken, mint a MC adathalmazon. Ez azt sugallja, hogy alkalmasabbak arra, hogy egy tesztszóhoz kiválasszák a leghasonlóbb szót egy listából, mint arra, hogy két szó pontos hasonlóságát megállapítsák.

Úgy gondoljuk, hogy a jövőben érdemes lenne módszereinket további, még nagyobb korpuszok segítségével kipróbálni, különösen a magyar verzió esetén (például Agirre et al. [10] egy 1,6 Terawordös angol korpuszt használtak, és algoritmusukat 2000 CPU magon futtatták). Továbbá mindenképpen szeretnénk a nyelvtani kapcsolatokat is alkalmazó modellt magyar nyelvre is implementálni, amivel reményeink szerint eredményeinket tovább tudnánk javítani. Ezen felül úgy véljük, mint azt a 2. fejezetben is említettük, hogy különböző típusú módszerek kombinálásával azok előnyeit ötvözhetjük. Ezért véleményünk szerint még jobb eredményeket tudnánk elérni, ha módszereinket kombinálnánk más, webes kereséseket vagy lexikális adatbázisokat felhasználó módszerekkel.

Hivatkozások

1. Dobó, A., Csirik, J.: Computing Semantic Similarity Using Large Static Corpora. In: van Emde Boas, P. et al. (eds.): SOFSEM 2013. LNCS, Vol. 7741. Springer, Heidelberg (2013, forthcoming) 491–502
2. Tsatsaronis, G., Varlamis, I., Vazirgiannis, M.: Text Relatedness Based on a Word Thesaurus. Journal of Artificial Intelligence Research, Vol. 37 (2010) 1–39
3. Higgins, D.: Which Statistics Reflect Semantics? Rethinking Synonymy and Word Similarity. In: Kepser, S., Reis, M. (eds.): Linguistic Evidence: Empirical, Theoretical and Computational Perspectives. Mouton de Gruyter, Berlin, New York (2005) 265–284
4. Rapp, R.: Word Sense Discovery Based on Sense Descriptor Dissimilarity. In: 9th Machine Translation Summit. Association for Machine Translation in the Americas, Stroudsburg (2003) 315–322

5. Turney, P.D., Littman, M.L., Bigham, J., Shnayder, V.: Combining Independent Modules to Solve Multiple-choice Synonym and Analogy Problems. In: 4th Conference on Recent Advances in Natural Language Processing. John Benjamins Publishers, Amsterdam (2003) 482–489
6. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review, Vol. 104 (1997) 211–240
7. Clark, S., Curran, J.R.: Parsing the WSJ using CCG and log-linear models. In: 42nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg (2004) 103–110
8. Lin, D.: An information-theoretic definition of similarity. In: 15th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1998) 296–304
9. Hughes, T., Ramage, D.: Lexical Semantic Relatedness with Random Graph Walks. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (2007) 581–589
10. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and WordNet-based approaches. In: 10th Annual Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies. Association for Computational Linguistics, Stroudsburg (2009) 19–27
11. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: 14th International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco (1995) 448–453
12. Patwardhan, S., Pedersen, T.: Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In: 11th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg (2006) 1–8
13. Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and Semantic Similarity. In: 4th Conference on Recent Advances in Natural Language Processing. John Benjamins Publishers, Amsterdam (2003) 212–219
14. Kulkarni, S., Caragea, D.: Computation of the Semantic Relatedness between Words using Concept Clouds. In: International Conference on Knowledge Discovery and Information Retrieval. INSTICC Press, Setubal (2009) 183–188
15. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: 20th International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco (2007) 1606–1611
16. Milne, D., Witten, I.H.: An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In: 23rd AAAI Conference on Artificial Intelligence. AAAI Press, Menlo Park (2008) 25–30
17. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: 15th international conference on World Wide Web. ACM Press, New York (2006) 377–386
18. Matveeva, I., Levow, G.-A., Farahat, A., Royer, C.: Term Representation with Generalized Latent Semantic Analysis. In: 5th Conference on Recent Advances in Natural Language Processing. John Benjamins Publishers, Amsterdam (2005) 45–54

A ReALIS tudástároló és következtető alrendszere

Kilián Imre

ReALIS ESzNyK / PTE TTK Informatika Tanszék
7624 Pécs, Ifjúság útja 6.
kilian@gamma.ttk.pte.hu

Kivonat: A ReALIS elemzési stratégiája sok ponton együttműködik egy háttérben meghúzódó tudáskezelő alrendszerrel. Ennek multimodális logikai keretben kell működnie: a ReALIS maga is erősen épít a többszereplős, episztemikus modellre, de – érthető módon – egy temporális logika következtető szabályrendszer megvalósítására is. Harmadikként – egy jogi alkalmazás kilátásai miatt – a deontikus modalitást is modellezzük. A cikk a fenti elméleti alapvetések Prolog leképezését tárgyalja, ami kiterjed a világocskaszerkezet, a modális logikai kifejezések és az ismert logikai axiómák megvalósítására, valamint beszámol a munka jelenlegi helyzetéről, és az első teszteredményekről is.

1 Bevezetés

A ReALIS elemzési stratégiája sok ponton egy tudáskezelő alrendszer létezését tételezi fel, és követeli meg. Egyfelől már a nyelvi elemzés is sok helyen csak egy háttérbeli tudástár alapján megválaszolható döntéseket igényel (pl. a „vörös ukrán szeszcsempész” jelzőinek sorrendjét egy „szín?”, ill. „nemzetiség?” információra vonatkozó kérdéssel lehet eldönteni). Másrészt a σ eventuális függvény összeállításához a lexikonbeli alaknak szintén a tudástárban tárolt elemekre kell hivatkoznia. Harmadrészt a σ függvény végeredménye alatt a közölni kívánt információ valamiféle logikai alakját értjük. Felmerülhet az a kérdés is, hogy egy teljes szöveget mennyire lehetséges csupán ebben a logikai alakban tárolni, és az is, hogy az ilyen alakú tárolás felett hogyan lehet keresési feladatokat futtatni. A logikai alakon emellett következtetési lépéseket lehet és kell végrehajtani. Ha kijelentés jellegű új értesülésről van szó, akkor arra vonatkozólag esetleg logikai konzisztenciavizsgálatot vagy következtetési lépéseket kell végezni, és végeredményben esetleg a belső tudástárban tárolni is kell. Ha kérdésről lenne szó, akkor viszont azt esetleg meg kell válaszolni.

2 Ontológiák és használati környezeteik

A fenti jellegű kérdéseket egy háttérbeli rögzített tudástár, egy ontológia alapján lehet megválaszolni. A ReALIS elemzéshez célszerű egy már létező ontológialeíró nyelvet választani, és egy kész ontológiát újrafelhasználni, esetleg a konkrét elemzési igé-

nyekkel kibővíteni. Erre a célra a Szemantikus Világháló projekthez kapcsolódva, annak OWL nyelvét választhatjuk, kiegészítve az SWRL szabályleíró résznyelvvel. Az előbbi a statikus taxonómiák leírására, az utóbbi a dinamikus viselkedések modellezésére alkalmas.

Az ontológiák használatát tekintve figyelemre méltó a Protégé ontológiaszerkesztő szoftver, amely egyébként szabad szoftver is. A Protégé korlátozott értelemben következtetési lépések végrehajtására is alkalmas, de ezeket egy alkalmazói szoftverhez külön kell megvalósítani, ill. meglevő következtető motorokat szabványos felületen keresztül meghívni.

A ℜeALIS egyértelműen Prologra építő működésmódja miatt a jelenleg használatos SWI-Prolog rendszer alatt futó Thea ontológia-kezelő csomag használata célszerű [7]. Ez némi hátrányt, de előnyt is rejteget magában. Egyrészt a modalitások kezelése miatt feltehetőleg az OWL Prolog tárgymodelljét kismértékben módosítani kell. Másrészt viszont az SWRL szabálynyelv Prologra történő testre szabása és/vagy Prolog alapú következtető motor létrehozása lehetővé teszi a szabályalkalmazói rendszer rugalmas változtathatóságát is.

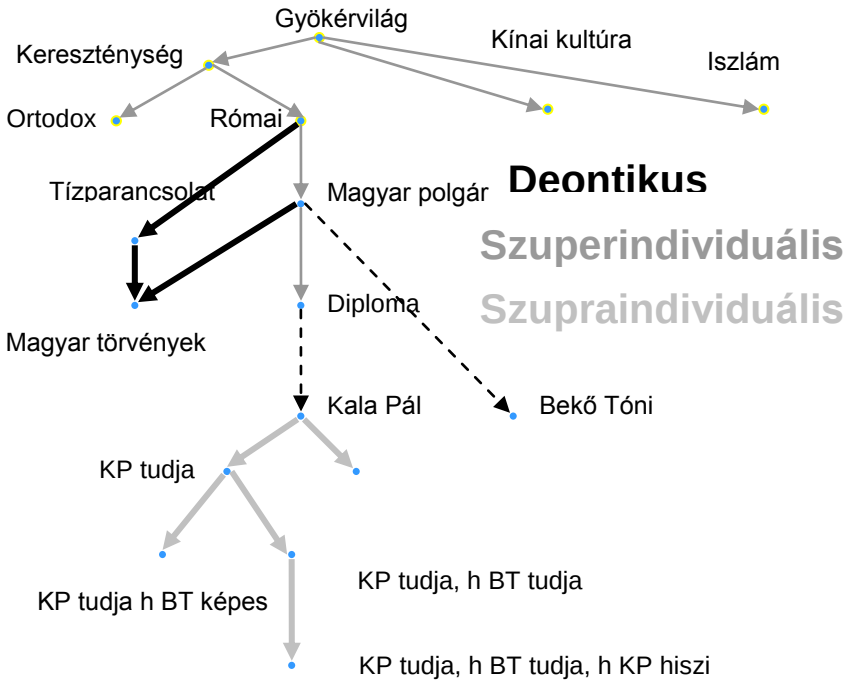
3 Modális világszerkezet és megvalósítása

A ℜeALIS alapvetően egy multimodális logikai rendszert tételez fel a háttérben. Az egyes interpretálók modellezése miatt többszereplős, episztemikusan modális rendszerre, az időviszonyok modellezése miatt temporálisan modális rendszerre van szükség. Továbbá – egy jogi alkalmazási lehetőség és környezet miatt – a rendszernek deontikusan is modálisnak kell lennie.

A multimodális rendszer egy fa, ill. körmentes háló topológiájú világszerkezetet tételez fel. Meghatározásunk nem követi a világszerkezet felett értelmezett közismert Kripke-féle szemantikát [6]. Ehelyett számba vesszük a használatos modális rendszerek axiómáit, és közülük a használni kívántakat a Prolog megvalósítási programnyelv eszközeivel megvalósítjuk.

A világok felett egy faszerkezetű viszony feszül ki, amelynek W_o a gyökéreleme, amely a megváltoztathatatlannak és félreérthetetlennek tekinthető külvilágot jelképezi, és az általánosan és megdönthetetlenül igaznak feltételezett tudáselemeket tárolja. A világ entitásainak egy részhalmaza a rendszer egyes ismeretszerzésre, nyelvek értelmezésére, valamint saját célkitűzések alapján önálló cselekvésre is képes szereplőinek (agents) A halmaza, amelyet első olvasatban a W_o objektív külvilág egy részhalmazaként értelmezhetünk. A rendszer faszerkezetét a W=W[i,t] függvény feszíti ki úgy, hogy egy w_1 világocskától az i cselekvővel és a t időjelölővel, valamint további modális címkékkel (hit, vágy, szándék, érzékelés, álom, stb.) címkézett élek mutatnak a másik, w_2 világocska felé. Az ilyen módon képzett újabb világocskák az episztemikus mélységnek felelnek meg (X /a bíró/ tudja, hogy Y /a tanú/ nem hiszi, hogy Z-nek /a vádlottnak/ szándékában állt volna pl. egy bűncselekmény elkövetése). A tetszőlegesen iterált mélység mélységet a rendszer paramétereként akár korlátozhatjuk is. A szereplőhalmaznak tehát a világszerkezet egy vágata felel meg: minden szereplőhöz

egy újabb részfa tartozik. A világszerkezet ilyen módon tetszőleges mélységben egymásba ágyazható, de a gyakorlatban ennek valamilyen korlátozása lehetséges.



1. ábra. A ℜeALIS modális világocskaszerkezete.

A fenti világocskarendszert a ℜeALIS mint modális logikai rendszer Kripke-féle keretszerkezetének is tekinthetjük, amelyet többszereplős esetben a c szereplő szerint szegmentálhatunk F_c=<W,R_c>. A világocskahalmaz legfőbb ismérve, hogy bennük az egyes elemi állítások *különbözőféleképpen is kiértékelődhetnek* (egyes cselekvők más és más személyes ismeretségi körrel rendelkeznek, mások tévesen úgy is gondolhatják, hogy az Anyám tyúkját Petőfi helyett Arany János írta, vagy hogy Magyarország fővárosa Bukarest, netán – ha török az illető – II. Nagy Szulimánt országépítő hősként tisztelheti, míg mi inkább zsarnok hódítónak tekintjük).

A világok feletti R elérhetőségi reláció megfelel a ℜeALIS W[i,t] függvényének, ami cselekvőfüggő is; egy adott szereplő nemigen tud közvetlen különbséget tenni egy társa által a múltban vagy a jövőben tudott vagy hitt dolgok között, másrészt egy adott, egészséges szereplőre nézve a tudott és hitt dolgok, valamint a valóság nemigen állhat ellentmondásban. Mindeközben a tudott és a kimondott dolgok ellentmondásban állhatnak – például ha célunk valaki megtévesztése, – az ilyet a világocskák faszerkezetének külön ágaiban tároljuk. A fent meghatározott fa-/körmentes gráf szerkezet

részekre bontható, amely megfelel az R relációhalmaz két részre, a RS és a RA halmazra történő szétválasztására. Az RS halmaz az *egyéniségek feletti (szuperindividuális) vagy csoport szintet* jelenti, és a csomópontjai az önálló tudásmennyiséggel rendelkező csoportokat jelzik. A RS relációhalmaz így szereplőfüggetlen, és egyfajta (az objektumorientáláshoz hasonló) öröklődési viszonyt értelmezhetünk rajta, azaz egy w_1 világocska nem ábrázolja közvetlenül, de igaznak tekinti az örökölt tudást is, a W_o gyökérvilág pedig a minden szereplő számára kézenfekvő és vitathatatlan ismereteket tárolja.

A csoporttudások szintjén az R relációhalmaz általában nem tiszta faszerkezetű: egy csoport, egy szellemi iskola általában több másik szellemi iskola eredményeire és tudására épít, egy szereplő pedig több csoporthoz tartozva, azok tudását összegezheti is. Ezért az RS részrelációra vonatkozóan az összefutó éleket is meg kell engednünk, vagyis a részreláció nem fát, hanem körmentes irányított gráfot (KIG) feszít ki.

Szintén itt helyezkedik el a normarendszert tároló deontikus világocskahalmaz is. Egy ilyen világocska valamely csoporthoz tartozó normarendszer ábrázolására használatos (pl. a MagyarTörvények világocskát a MagyarPolgárok csoportból vezetjük le. A normák egymásra is épülhetnek, és közöttük is öröklődés értelmezhető. Például valamely helyi városi rendelethalmaz hatókörén belül az örökölt MagyarTörvények is érvényesek.

Az RA részrelációt az *egyéni alatti (szupraindividuális)* szintnek nevezzük, amely a szereplők hitére, meggyőződésére stb. vonatkozó információkat tárolja. Ez az R és az RS relációk különbsége, topológiailag egy erdő, melynek gyökérpontjai az egyes egyedi szereplőket (individuumokat) jelölik.

3.1 A ℜeALM tudásleíró nyelv és a tudástár

A ℜeALM nyelv az elemzés végeredményét, a leírt mondatok logikai alakját rögzíti, és ilyen értelemben egy szöveg belső, logikai ábrázolási formájának is tekinthető. A nyelv tükrözi a már említett modális logikai leíró képességet. A modális operátorokat az alábbi négyesekből álló (csak részlegesen rögzített) halmaz írja le:

M={<L={bel,des,int,…},GR={min,med,max},A,T,P={+,0,-}>}, ahol:

- L (bel*ieve*, des*ire*, int*ent*): a multimodalitásért is felelős *modális címke*. Ez az igényeknek és a nyelvtani elemzésnek megfelelően tovább bővíthető, pl. a retorikai relációkkal (*supp, cons,* stb.**)**, vagy az egyes érzékelésfajtákat leíró *hear, smell, taste, touch* címkékkel.
- GR (minimal, medium, maximal): a *modalitás fokozata*. A <bel,min,…> jegy jelzi a hagyományos **B** (believe / gyenge episztemikus) operátort. A **K** (know) operátort a <bel,max,…> párral adhatjuk meg.
- T *időjelölő*, ami jelenthet egyszerű időpontot, időintervallumot, vagy bármilyen bonyolultabb időmeghatározást. Az időjelölőkön végezhető műveleteket közvetlenül Prolog szinten valósítjuk meg.

- A *szereplők halmaza*. A szereplők megfelelnek a háttérontológia egy osztályának. Az RS relációban résztvevők közbülső elemei aktív társadalmi csoportoknak (Society), a levélelemek, valamint az RA relációban résztvevők gyökérelemei aktív egyéneknek (Agent / HumanAgent) felelnek meg.
- P={+,-} pozitív vagy negatív *modális polaritás*: amely a negatív modális operátor leírására szolgál, a gyakorlatban a *konstruktív tagadás* számára használjuk.

A ℜeALM nyelv végeredményben az elsőrendű logika (ill. Prolog) modális kiterjesztését adja úgy, hogy minden logikai kifejezés (állítás vagy literál) modális világocskába helyezhető, amit a `MOD:EXPR` szerkezettel fejezhetünk ki. A többszörös mélységű egymásba ágyazást azonban nem a `:/2` funktor iterálásával, hanem a modális címkék Prolog listába foglalásával fejezzük ki. Az alábbi reprezentáció pl. egy vádlott alibijének a bíró elméjében történő ábrázolását mutatja be (a vádlott egy Fradi–Újpest meccsen lett volna, amely a bűntény helyszínétől 100 km-re volt).

```
[bel(bíró,max,+),tell(vádlott)]:
     (bűntény(X,LOC,TIME), tartózkodik(vádlott, M),
     futballmeccs(M, fradi, újpest,L,T),
     TIME⊆T, táv(LOC,L,100km)).
```

3.2 Az episztemikus modalitás megvalósítása

A modalitások logikába, így Prologba is történő leképezésére Ohlbach ad cikkében javaslatot [5]. Ezt követve egy elsőrendű logikában felírt `p` literális kifejezést első paraméterként kibővítünk egy MOD modális kifejezéssel, vagyis

`p(X1,X2,…,Xn)` → `p(MOD,X1,X2,…,Xn)`

A `MOD` kifejezés a literál modális környezetét (a modális világocskát) írja le, amely a ℜeALM résznyelv hasonló kifejezéseit követi. Eszerint `MOD` a következő formákat öltheti:

`root`	megfelel a gyökérvilágnak
`ID`	ahol `ID` egy Prolog azonosító (logikai konstans), egyes csoportok azonosítóit jelöli, amelyek megfelelnek a háttér-ontológia „társadalmi csoport” fogalmának.
`root(ID)`	ahol `ID` egy Prolog azonosító (logikai konstans), megfelel az egyes egyéni szereplők azonosítóinak. Amennyiben a modális címke maga a szereplőazonosító, akkor ezzel az adott szereplő gyökérvilágát jelöljük.

`bel(GR,I,T,P,F)`
`int(GR,I,T,P,F)`
…
	az egyes modalitásfajtákhoz külön modális címkeazonosító (Prolog függvényszimbólum) tartozik, ahol GR (grade) a modalitás fokozata, I a szereplőazonosító, T az időjelölő, F pedig a szülő világocska modális kifejezése

`GR`	A modalitás fokozatot egész számokká képezzük le (pl. 0,1,2). Így esetlegesen számtani műveleteket, (pl. összehasonlítást) tudunk rajta végezni.

A világok feletti relációt a `world/2` Prolog állítás rögzíti. Ez két részből áll. A szuperindividuális szinten az egyes csoportok azonosítóira hivatkozik, amelyet az `sWorld/2` reláció tényállításonként tárol. A reláció tartalma esetlegesen egyes ontológiabéli relációkkal is kifejezhető (melyik szellemi iskola milyen másik csoporttudásra épít). A szupraindividuális szinten reláció a modális címkékből kifejezhető, pl. az alábbi, vagy hasonló Prolog kód segítségével:

```
iWorld(SUP,bel(GR,I,T,P,SUP)).
```

A világocskarelációk rögzítésének legfontosabb célja a felettük megvalósítandó öröklési műveletek bemutatása. Első közelítésben mindenütt csak monoton öröklést tételezünk fel, azaz a leszármaztatott világocskák tudása csak nőhet az ősvilág tudásához képest. A következő öröklési viszonyokat valósítjuk meg:

1. Mód nélküli gyökérontológia elérése: A gyökértudásban valamiféle közös ismerethalmazt ábrázolunk, ami célszerűen egy már létező ontológiából származhat, amely azonban nem modális. Ezért itt a mód nélküli ontológia definícióira történő egyszerű visszavezetést tároljuk, amelyet minden definícióra meg kell adnunk. A mód nélküli ontológia az ontológia rövid álnevének megfelelő Prolog modulba kerül (pl. `sumo`), míg a modális ontológia egyetlen, önálló modulban foglal helyet.

   ```
   mammal(MOD,X):-sumo:mammal(X).
   ```

2. Szuperindividuális öröklés: Minden definícióra megadjuk, hogy a közvetlen ősvilágocskából örökölhet.

   ```
   p(SUB,X1,X2,…,Xn):-
     sWorld(SUP,SUB), p(SUP,X1,X2,…,Xn).
   ```

   Ez a megoldás sajnos csak akkor alkalmazható, ha a csoportszerkezet topológiája szigorúan fa. Ha mégsem, akkor ősöket előbb összegyűjtjük a következőképpen:

```
p(ID,X1,X2,…,Xn):-
  sAncestors(ANC,ID), member(SUP,ANC),
  p(SUP,X1,X2,…,Xn).
```

3. Szupraindividuális fokozatöröklés: Az egyének episztemikus világocskáiban az öröklés egyáltalán nem kézenfekvő (nem biztos, hogy valaki tudja is azt, ami egyébként igaz). Érvényes viszont a fokozatok közötti öröklés axiómája, vagyis, ha valaki tud valamit, akkor hiszi és sejti is ugyanazt. Ez szintén minden lehetséges definícióra a következő állítással fejezhető ki:

```
p(bel(GR1,I,T,P,F),X1,X2,…,Xn):-
  p(bel(GR2,I,T,P,F),X1,X2,…,Xn), {G1<G2}.
```

(A `{}/1` hívás a CLPR racionális megoldót hívja: a gyakorlati jelentősége, hogy csak akkor értékelődik ki, ha mindkét oldala kiértékelődött.)

Az egyszerű öröklési axiómák mellett tekintsük át az elutasított modális axiómákat is.

1. A Kripke-féle **K** axióma (omniscience problem) egy alapvető kérdést vet fel. Igaz-e, hogy egy implikációs szabály és az implikáció feltételének ismeretében a következmény is ismert? Igaz-e mindez tranzitíven hosszú következtetési láncokra is? Ezek miatt az axióma elvetését, esetleg legfeljebb valamilyen korlátozott láncon való megvalósítását javasoljuk.
2. Még kevésbé látjuk használhatónak az igazolhatóság (**T**) axiómáját. Eszerint, ha valamit tudunk, akkor az úgy is van. A szubjektív tudásból semmiképpen sem következtethetünk az objektív világra.
3. A pozitív önismeret (**4**) axiómája szerint, ha tudunk valamit, akkor azt is tudjuk, hogy tudjuk. A negatív önismeret axiómája szerint (**5**) viszont tudjuk azt is, amit nem tudunk. Ezek megvalósítása az imént vázolt rendszerben a tudáselemek eggyel mélyebbi modális szinten történő megismétlését jelenti. Ezért az axiómát korlátozottan, vagy egyáltalán nem alkalmazzuk, hiszen ez az összes szereplő esetében mind a hiányzó, mind ismeretek többszörös tényszerű tárolását követelné meg. Sőt, a művelet matematikailag korrekt lezárása esetén végtelen adatmennyiségeket jelentene (nemcsak azt tudjuk, hogy mit tudunk, hanem még mindezt is tudjuk, sőt…).

3.3 Az időmodalitás kezelése

A ℜeALIS rendszer az időmodalitást némileg korlátozott formában tételezi fel. Bár a <L,GR,i,**before(T)**>, ill. a <L,GR,i,**after(T)**> modális címkék nyilvánvalóan valamiféle időmodalitást írnak le, a használatuk több korlátozást is jelent:

Ahhoz, hogy az időcímke kifejezze, hogy a címkézett állítás a jelölt időintervallumban folyamatosan fennállt-e (*erős modalitás*), vagy az időintervallum egy részében áll-e fenn (*gyenge modalitás*), a fenti címkéket gyenge modalitásként értelmezzük, és

bevezetjük az erős modalitást kifejező beforeA és afterA (before/after Always) címkéket is.

1. Múlt-jövő szimmetriája: az axióma a gyakorlati szempontból csupán annyit jelent, hogy minden, a múltra vonatkozó definíciót a jövőre vonatkozólag is megismétlünk. Az alábbiakban általában csak az egyik irányú axiómákat mutatjuk be, amelyet tehát az időtengelyre nézve értelemszerűen tükrözni kell.
2. Dualitás: Ha valami a múltban valaha igaz volt, akkor az ellenkezője nem állhatott fenn folyamatosan. Az axióma Prolog nyelven könnyedén leírható, azonban a Negation As Failure (nem konstruktív negáció) miatt csak igen korlátozottan működőképes.

   ```
   p(before(T),x1,…,xn):-
     \+ p(\+afterA(T),x1,…,xn).
   ```

   Az efféle jellegű következtetésekre a Prolog önmagában képtelen, ha ilyenre tényleg szükségünk van, akkor a következtetési mechanizmust ki kell terjesztenünk.
3. Időbeli általánosítás axiómája (**TG**, Temporal Generalization): Ha valami időfüggetlenül igaz, akkor igaz a múlban és a jövőben is. Az axióma könnyen megvalósítható az alábbi Prolog szabállyal:

   ```
   p(before(T),x1,…,xn):- p(x1,…,xn).
   ```

A TL0, TL1 és TL2 rendszer axiómáihoz a modális időcímkék iterációja (egymásba ágyazhatósága) szükséges.

1. A TL0 rendszer antiszimmetria axiómája (**antisym**) kimondja, hogy ha valami időfüggetlenül igaz, akkor mindig igaz volt, hogy valamikor igaz lesz. Ennek Prolog ábrázolása:

   ```
   p(after(beforeA(T)),x1,…,xn):- p(x1,…,xn).
   ```

2. A TL1 tranzitivitási axiómája (**tra**) szerint ha valami mindig igaz lesz, akkor mindig igaz lesz az is, hogy mindig igaz is marad. Vagyis a (jövőbeli) erős modalitás tetszőlegesen iterálható.

   ```
   p(afterA(afterA(T)),x1,…,xn):-
   p(afterA(T),x1,…,xn).
   ```

3. A TL2 **trichotómia** axiómája kizárja a párhuzamos idősíkok létezését. Eszerint, ha a jövőben valamikor vagy A vagy B megvalósul, akkor vagy mindkettő egyszerre valósul meg, vagy az egyik a másik után később, vagy fordítva. Az axióma a jelen rendszerben nem valósítható meg kézenfekvő módon.

4. A TL5 sűrűségaxiómája (**den**) az idősíkot tetszőlegesen sűrűvé teszi azzal, hogy kimondja: ha valami valamikor igaz lesz, akkor lesz valamikor egy

közbülső pillanat is, amikor úgy lesz, hogy valamikor igaz lesz. Az axióma Prolog ábrázolása:

```
p(after(after(T)),x1,…,xn):- p(after(T),x1,…,xn).
```

3.4 Példa az episztemikus modális tudásábrázolásra

Károly [2] az általa elkészített példában az „Egye fene, elmehetsz!” mondatot úgy értelmezi, mint amikor a társ erős vágyat érez a távozásra, ám ezt a beszélő eleinte még nem akarja, de később gyengén szándékba veszi. A hivatkozott cikkben látható grafikus ábrának a következő Prolog tudásbázis felel meg:

Az alábbi két állítás az OWL tudásbázis Prolog fordításából származik. Terminológiai információt ír le (T-Box): a folyamat, a mozgás és a gyaloglás öröklődését.

```
process(MOD,PR):- motion(MOD,PR).
motion(MOD,PR):- walking(MOD,PR).
```

A w1 gyaloglás-esemény szereplője a1, akinek erős vágya, hogy elmenjen.

```
agent(w1,a1).
walking(des(max,a1,present,+,root(a1)),w1).
```

Sajnos azonban a w2 eseményt az a1 szereplőjével, vagyis a távozást a2 egészen idáig nemigen akarta.

```
agent(w2,a1).
walking(des(med,a1,past,-,root(a2)),w2).
```

Végül azonban mégis, gyengén szándékba veszi:

```
agent(w2,a1).
walking(int(min,a1,present,+,root(a2)),w3).
```

A fenti tudásbázis tesztelhető például a következő kérdésekkel:

1. Mit akart a1 korábban? A másodrendű kérdés úgy reifikálható első rendbe, hogy „volt-e a1 ágensű, erősen vágyott folyamat a múltban”

```
?-process(des(max,a1,TIME,+,root(a1)),W),
  earlier(TIME,present).
```

2. Mit szándékszik a1?

```
?-process(int(GRADE,a1,present,+,root(a1)),W).
```

3. Ki változtatta meg a korábbi véleményét egy másik szándékra? A kérdés megint másodrendű, amely reifikálva ismét folyamatokra kérdez.

```
?-process(des(_,WHO1,TIME,-,root(WHO)),W1),
   earlier(TIME,present),
   process(int(_,WHO1,present,+,root(WHO)),W2).
```

4 Eredmények, továbbfejlesztés

A fenti leírás során a rendszer megvalósítását megelőző deszkamodell-tanulmányprogramok eredményeire támaszkodtunk. Az eredmények rendszerbe szervezése, a nagyméretű ontológiák beintegrálhatóságának és cserélhetőségének biztosítása, tesztforgatókönyvek létrehozása folyamatban van.

Mindezek időt és energiát kérnek. A továbbfejlesztésnek azonban van egy sokkal érdekesebb iránya is. Nevezetesen: a következtetési stratégiák kérdésében a szerző meggyőződése, hogy a rendszer ki kell egészíteni valamilyen metaprogramozási nyelvvel és környezettel, amely a hallott információt az értelmező és a beszélő habitusának megfelelő stratégia szerint tárolja el a modális világocskarendszer mélyebb vagy magasabb modális szintjein.

A szerzőt e cikk alapjait jelentő kutatásaiban a TÁMOP-4.2.1.B-10/2/KONV/2010/KONV-2010-0002 (A Dél-dunántúli régió egyetemi versenyképességének fejlesztése) projekt támogatta.

Itt szeretnék köszönetet mondani a ℜeALIS projektbéli munkatársaimnak, Alberti Gábornak, Kleiber Juditnak és Károly Mártonnak a nyelvészeti információk önzetlen átadásáért és a jól célzott, és egyben megfelelően adagolt, a cikk végső példányára is kiható megjegyzéseikért.

Hivatkozások

1. Alberti G.: ℜeALIS. Interpretálók a világban, világok az interpretálóban. Akadémiai Kiadó, Budapest (2011)
2. Károly M.: Interpretáció, intenzionalitás, modalitás – avagy a ℜeALIS λ függvényének interpretációja felé. In: VIII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2011)
3. Niles, I., Pease, A.: Origins of the Standard Upper Merged Ontology: A Proposal for the IEEE Standard Upper Ontology. In: Proceedings of Measuring Intelligence and Performance of Intelligent Systems Conference (2001)
4. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In: Proceedings of the Twelfth International World Wide Web Conference. ACM (2003) 48–57
5. Ohlbach, H.-J.: A Resolution Calculus for Modal Logic. FB Informatik, University of Kaiserslautern, Germany (1988) (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.5003, letöltve: 2012. június 25.)
6. Ruzsa I.: Klasszikus, modális és intenzionális logika. Akadémiai Kiadó (1984)

7. Vassiliadis, V., Wielemaker, J., Mungall, C.: Processing OWL2 ontologies using Thea: An application of logic programming. In: Hoekstra, R., Patel-Schneider, P.F. (eds.): Proceedings of OWL: Experiences and Directions (OWLED), CEUR Workshop Proceedings, Vol. 529 (2009) (http://www.webont.org/owled/2009, letöltve: 2012. június 25.)

Az igazság pillanata – avagy a ℜeALIS α horgonyzó függvénye

Alberti Gábor, Károly Márton, Kilián Imre, Kleiber Judit, Vadász Noémi

PTE BTK Nyelvtudományi Tanszék
ℜeALIS Elméleti és Számítógépes Nyelvészeti Kutatócsoport
7624 Pécs, Ifjúság útja 6.
alberti.gabor@pte.hu, harczymarczy@gmail.com, kilian@gamma.ttk.pte.hu, kleiber.judit@pte.hu, vadasznoemi@gmail.com

Kivonat: Kutatócsoportunk szeme előtt változatlanul az a hosszú távon kifizetődő cél lebeg, miszerint az intelligens számítógépes nyelvészeti célokat (pl. fordítás, tudásreprezentáció, jelentés-egyértelműsítés) az egymással kommunikáló humán interpretálói „elmék” ℜeALIS-modelljének implementálására alapozva kívánjuk megvalósítani [1–4, 7–8]. Idén a diskurzusreferensek [4] azonosításáért felelős α függvényt vesszük górcső alá, rámutatva, hogy a komolyabb kihívást jelentő jelentéskapcsolatok [14, 18] megragadása számára a ℜeALIS dinamikus és reprezentacionalista jelentésmegközelítése a lehető legtermészetesebb közeg. Majd felvázoljuk az elméleti hátterét a Károly Márton által demonstrált [13] igazságértékelő programnak, amely az alapmodulja annak a programcsomagnak, amelynek előmunkálatairól 2010-ben és 2011-ben beszámoltunk [6, 12, 15].

Kulcsszavak: igazságértékelés, modellelméleti szemantika, intenzionalitás

1. A kopredikáció

A ℜeALIS referens-felfogását Landman nyelvfilozófiai elméletére [16] alapozzuk: a referensek kezdetben üres, belső „fogasok” az emberi elmében, melyekre a külvilág észlelésének hatására egyre több információ települ.[1]

A ℜeALIS (lényegében *reifikációt* alkalmazva) úgy ragadja meg (σ függvényének segítségével) egy hagyományos predikátumlogikai formula tartalmát, hogy egy referens-fogashoz rendeli annak valamennyi komponensét, egy-egy referensnek tekintve a predikátumot éppúgy, mint az argumentumokat (l. (1c) alább). Egy szöveg tartalma ezek után az így reprezentált formulák összesítéseképpen úgy ragadható meg, hogy összehorgonyozzuk a *kopredikáló* [1, 2, 7] – ugyanarra utaló – referenseket, a grammatikai relációk nyújtotta kulcsokra támaszkodva (1d).

[1] Részletes ismertetéssel és összevetéssel szolgál: [4].

1. példa. KOPREDIKÁCIÓ – IDEALIZÁLT HELYZETBEN

a. A legjobb barátom röviddel az egyetem után belefoghatott a szerintem legjobb pécsi házába.

b.
e1: p1 t1 r11	*legjobb*
e2: p2 t2 r21 r22	*barátja*
e3: p3 t3 r31	*rövid*
e4: p4 t4 r41	*egyetem*
e5: p5 t5 r51 r52 r53	*után*
e6: p6 t6 r61 r62	*belefog*
e7: p7 t7 r71 r72	*-hAt*
e8: p8 t8 r81 r82	*szerinte*
e9: p9 t9 r91	*legjobb*
e10: p10 t10 r101	*pécsi*
e11: p11 t11 r111 r112	*háza*

c.
$\sigma : \langle Pred,e11\rangle \mapsto p11$
$\sigma : \langle Temp,e11\rangle \mapsto t11$
$\sigma : \langle Arg1,e11\rangle \mapsto r111$
$\sigma : \langle Arg2,e11\rangle \mapsto r112$

d.
$\alpha : \langle ...,r11\rangle \mapsto r21$	$\alpha : \langle ...,p1\rangle \mapsto p_{legjobban\_szeretett}$
$\alpha : \langle ...,r22\rangle \mapsto r_{én}$	$\alpha : \langle ...,p2\rangle \mapsto p_{barátja}$
$\alpha : \langle ...,r31\rangle \mapsto r53$	$\alpha : \langle ...,p3\rangle \mapsto p_{néhány\_hónap}$
$\alpha : \langle ...,r52\rangle \mapsto r41$	$\alpha : \langle ...,p4\rangle \mapsto p_{egyetemi\_évek}$
$\alpha : \langle ...,r51\rangle \mapsto e7$	$\alpha : \langle ...,p5\rangle \mapsto p_{utána\_időben}$
$\alpha : \langle ...,r61\rangle \mapsto r21$	$\alpha : \langle ...,p6\rangle \mapsto p_{belefog}$
$\alpha : \langle ...,r62\rangle \mapsto r111$	
$\alpha : \langle ...,r71\rangle \mapsto r61$	$\alpha : \langle ...,p7\rangle \mapsto p_{lehetőség\_vkinek}$
$\alpha : \langle ...,r72\rangle \mapsto e6$	
$\alpha : \langle ...,r81\rangle \mapsto r_{én}$	$\alpha : \langle ...,p8\rangle \mapsto p_{úgy\_gondol}$
$\alpha : \langle ...,r82\rangle \mapsto e9$	
$\alpha : \langle ...,r91\rangle \mapsto r111$	$\alpha : \langle ...,p9\rangle \mapsto p_{díjnyertes\_tervvel}$
$\alpha : \langle ...,r101\rangle \mapsto r111$	$\alpha : \langle ...,p10\rangle \mapsto p_{Pécsett\_lévő}$
$\alpha : \langle ...,r112\rangle \mapsto r21$	$\alpha : \langle ...,p11\rangle \mapsto p_{általa\_tervezett\_ház}$

A fenti (1a) mondat *után* szava például – hogy sok referenst hozó eventualitással (ezúttal egy állapot leírásával) kezdjük – azt vált(hat)ja ki, hogy egy e5 referenshez (l. (1b)) a sorában álló további referenseket az alábbit kifejezve rendeli hozzá a σ függvény: „az r51 helyzet az r52 időszak után r53 idővel áll be”. A szórend és a morfológia adta kulcsok alapján az α függvény az (1d) megfelelő soraiban közölt következő azonosító összehorgonyzásokért felelős: r51 egy előálló lehetőség (egy munka elkezdésére), r52 egy egyetem elvégzésének időszakával azonosítandó, amihez képest az r53 időszakról az derül ki, hogy „rövid”.

A „barát”-ról (r21) a morfológia és a szórend alapján kiderül, hogy az „enyém” (r22), és hogy róla állítjuk, hogy a „legjobb” (r11), legalábbis így (e9) gondolja a

beszélő (r81); továbbá amikor eljutunk az igéhez, akkor az is kiderül, hogy ő (r61) foghatott bele valamibe. Mibe (r62)? Egy házba (r111), amiről a jelzők azt mondják, hogy a „legjobb" (r91) és „pécsi" (r101), antecedenskeresés útján pedig az adódik, hogy a „baráté" (r21).

Hogy miféle cselekvésre állott elő a *-hAt* morféma kifejezte lehetőség (r72), azt a kötött morfémát megkötő igető árulja el: belefogni valamibe (e6); ez a morfológiai helyzet pedig egyben azt is meghatározza, hogy a cselekvő (r61) egybeesik a lehetőség kedvezményezettjével (r71).

A kopredikációnak ezt az idealizált megvalósítását már korábbi cikkekben is ismertettük [1, 2, 7]; a jelen mondat azonban olyan „orvosi ló" gyanánt szolgál, ahol a grammatikai relációkból közvetlenül kiolvasható *interszektív* jelentéskalkuláció [14] többnyire nem működne a gyakorlatban.

Tekintsük át először, hogy hol működik! A *pécsi ház* jelzős szerkezet esetében például működik azon ésszerű feltételezés mellett, hogy a „Pécsett található" dolgok és a „házak" halmazának *metszete* adja ki a jelzős szerkezet által leírt dolgot tartalmazó halmazt.[2]

A *pécsi vonat* jelzős szerkezettel azonban már gondja adódna az iménti interszektív kalkulációnak, mivel egy vonatot éppen olyankor nevezhetünk meg így, amikor az nem Pécsett található, hanem Budapesten, Szombathelyen vagy a Balaton partján. A megnevezés alapja ilyenkor az, hogy Pécsről jön vagy Pécsre tart.

2. A kopredikáció megmentése

Baj van a kopredikáció elvével?! Azt állítjuk, hogy nem – amennyiben a *vonat* szóról valamilyen módon feltételezni tudjuk, hogy előhív számunkra egy olyan referenst, amiről már állítható, hogy „Pécsett található". A Laczkó által [17] fogalmi keretként emlegetett gazdagabb lexikai jelentésreprezentációba például beleérthető a vonat kapcsán annak indító és végállomása. Ha ezeket implicit referenseknek tekintjük, akkor az így kiterjesztett argumentumszerkezet már az interszektív módszerrel is produkálni képes a megfelelő jelentésvariánsokat – beleértve azt a harmadikat is, hogy egy vonat tényleg mindig egyetlen városban található (ilyen *pekingi vonatot* például el tudunk képzelni).

A ℜeALIS elmereprezentáción alapuló dinamikus szemantikai [9–11] megközelítése kézenfekvően kínálja azt a megoldást, hogy egy predikátum számára keressünk horgonyozható referenst a predikátum egy korábbi előfordulásával összetársult predikátumok referensei között. Egy vonat kapcsán például gyakran elhangzik, hogy honnan indult vagy hová tartott; és ez a ℜeALIS élethossziglani rendszerében regisztrálódik (az L a *lifelong* rövidítése).

Az alábbi 2. példa azt szemlélteti, hogy a vizsgált mondat esetében hogyan működik a kopredikáció felvázolt általánosítása / kiterjesztése.

---

[2] Felhívjuk a figyelmet a kopredikáció két különleges esetére: az r72 és r82 referensek eventuális referensekhez horgonyzódnak („a cselekvés, hogy valaki belefog valamibe", „az állapot, miszerint valaki a legjobb"). E referenstípus bevezetésével tehát ilyenkor is két (1b) pontbeli formula „metszeteként" foghatjuk fel egy grammatikai reláció hatását.

Az *után* r52 referensét például akkor horgonyozhatjuk össze az *egyetem* valamely referensével, ha az időtartamot mutat. Hogyan juthatunk el „a legjobb barát egyetemi éveihez”, amihez az idealizált 1. példában ? Az élethossziglani információhalmozás során egy ℜeALIS típusú rendszerbe szükségszerűen bekerül az egyetemi forgatókönyv megannyi eleme: az oktató és a hallgató, és ezzel együtt azok napok rutinja az órarendhez kötődően, illetve években mérhető teljes egyetemi pályafutásuk. Ezek már megfelelő típusú, azaz időtartamot jelentő referensek. Pillanatnyilag annyit fogadjon el az olvasó, hogy el *lehet* jutni az imént említett szándékolt jelentéshez: a barát egyetemi évei tartamához.

Egy mondat jelentésreprezentációja tehát olyan hálózatbővítést jelent a σ (szavak által meghívott formulák), majd az α kiterjesztése révén (referensazonosítás a grammatika alapján), ami gyakran a σ további tartománybővítését igényli („áthidalás”) – éppen korábban kiépült α-kapcsolatok alapján (lexikális / kulturális / interperszonális tudás).

Ami a formalizmust illeti, a 2. példa 3. sorában azt írtuk fel, hogy az interpretálói elme információállapotában K4 darab α-horgonyzási kapcsolatélen átlépkedve jutottunk el ahhoz az r_{n4} referenshez, ami időtartam típusú, és éppen egy iskola elvégzésének időtartamát jelöli. Ezt a σ függvény értelmezési tartományának (1c) több rendbeli kiterjesztésével ragadhatjuk meg: a K4 rendbeli kiterjesztés értelmezési tartománya már olyan „fogalmi keretet” feszít ki az *egyetem* mint predikátum köré, amelyben az „elvégzésének időtartama” is ott van implicit argumentumként.

2. példa. KOPREDIKÁCIÓ – ÁTHIDALÁSSAL (σ FÜGGVÉNYT KITERJESZTVE α MENTÉN)

$\alpha : \langle ...,r11\rangle \mapsto \sigma(\langle K2,n2\rangle,e2)$	*milyen tekintetben legjobb?*
$\alpha : \langle ...,r31\rangle \mapsto \sigma(\langle K4,n4\rangle,e4)$	*mihez képest rövid?*
$\alpha : \langle ...,r52\rangle \mapsto \sigma(\langle K4,n4\rangle,e4)$	*milyen időszak után?*
$\alpha : \langle ...,r62\rangle \mapsto \sigma(\langle K11,n11\rangle,e11)$	*milyen tevékenységbe fogott bele?*
$\alpha : \langle ...,r91\rangle \mapsto \sigma(\langle K11,n11\rangle,e11)$	*milyen tekintetben legjobb?*
$\alpha : \langle ...,r112\rangle \mapsto \sigma(\langle K11,n11\rangle,e11)$	*milyen tevékenység alapján a háza?*

A σ függvény iménti kiterjesztése egy mondatelemzési folyamatban nemcsak egy lehetséges jelentés detektálását jelenti, de egyben olyan *elköteleződést* is jelent, ami más szavak jelentés-egyértelműsítését előmozdítja. Az r31 referens „rövid” volta például egy egyetemi hallgatói pálya 3–6 évéhez mérendő, amihez képest *néhány hónap* rövidnek számít (l. mindkét példát). Ugyanennyi idő nem számítana rövidnek egy náthából való felépülés esetében. Ezt az eddigiekben tárgyalt eszköztárral úgy ragadhatjuk meg, hogy az r31 implicit argumentum társargumentumává „fogadunk” egy r3' referenst a σ értelmezésitartomány-bővítése révén ilyen összefüggéssel: az r31 hossz (mondjuk) 10–40%-a az r3' hossznak; majd r3'-t egy olyan r5' implicit referenssel horgonyozzuk össze, amely egy egyetem elvégzésének tipikus hossza.

Ami a házba való „belefogást” illeti (r52), típusütközést gyaníthatunk itt is: belefogni egy cselekvésbe lehet. A házhoz kapcsolódó e11 eventualitáshoz a σ értelmezési tartományának a kiterjesztésével kell tehát értéket rendelni, mégpedig cselekvés típu-

sút. A Pustejovskyt [18] ismertté tevő összefüggéseket alkalmazva ilyen cselekvések például a létrehozás részét képezők: a ház megtervezése, felépítése, bebútorozása...

Amennyiben elkötelezzük magunkat a mellett a jelentés mellett, hogy a „barát" a ház megtervezésébe fogott bele, akkor ezzel olyan elköteleződést tételezünk fel, ami a *háza* morfológiájában megmutatkozó birtoklást is specifikálja, illetve a *legjobb* jelzői értelmezését. Keressük lényegében az r112-höz és az r91-hez horgonyozható referenseket. A birtokosi alak (*háza*) a tőhöz képest (*ház*) olyan explicit argumentumhalmazbővülést mutat, amelyben az új argumentumot a σ értelmezési tartományának a kiterjesztésével „foghatjuk be" – és preferáltan éppen az iménti kiterjesztéssel. Vagyis a „háza" éppen „az általa tervezett ház"-at fogja jelenteni.

Kifejtés nélkül jelezzük, hogy a *ház* jelzője, a *legjobb* preferáltan ugyanebben a jelentéskörben értelmezendő (r91), tehát olyasmire utalhat, hogy a tervezés dicsérhető. A *szerintem*-ben rejlő szubjektivitás azonban olyan interperszonális tudást is behozhat a beszélővel kapcsolatosan, ami mást jelentést specifikál.

Nyilván az (1a) mondat második szava – ugyancsak *legjobb* – a barát predikátum argumentumkörének a kiterjesztése révén nyer specifikált jelentést. Amennyiben a barát fogalmát az iránta érzett szeretet / ragaszkodás / bizalom felől határozzuk meg, akkor a jelző ezen érzelmek (valamelyikének) különleges mértékére utal.

A kopredikáció (címben említett) megmentésének jelentősége abban áll, hogy ha a grammatikai relációk mögött nem tudjuk kimutatni azt, hogy a grammatikai reláció két oldalán álló két kifejezés ugyanarról a szereplőről tesz állítást, akkor megfoghatatlanná válik a szavakra „bízott" jelentéselemek összegződésének a mikéntje. Az 1. szakaszban bemutatott triviális interszekció azonban csak akkor lenne lehetséges – amint arra Pustejovsky rámutat [18], ha a *legjobb*, *rövid*, *háza*, *belefog* típusú szavaknak több tucatnyi (vagy akár korlátlanul sok) konkrét jelentése lenne.

A ℜeALIS dinamikus reprezentacionalizmusa megadja a megoldás kulcsát. Mivel dinamikus, ezért a mondatelemzés során lehet előállítani a specifikált aktuális jelentéseket aktiválva korábban felépült jelentésstruktúrákat – így egy szóhoz akár korlátlanul sok jelentés is társulhat (újabb és újabb kontextusokban elindítva a kalkulációs folyamatot). Az élethossziglani reprezentáció pedig azt a relációs hálózatot szolgáltatja, ami minden egyes benne lévő predikátum (-referens) mint „középpont" felől nézve a jelentést a struktúrában elfoglalt viszonylagos pozícióként definiálja, hűen Saussure szelleméhez.

Hogy a kopredikáció teljes általánosságban megmenthető, az talán azzal a példával igazolható, amely megítélésünk szerint a legsúlyosabb kihívást jelenti: „Hozd már ide azt a rohadt/hülye/... létrát!" A jelző vonatkozhat a létra gyenge állapotára is (ez könnyen kezelhető eset), azonban vonatkozhat a címzettre is, vagyis arra, hogy a beszélő csúnyákat gondol róla.

Az utóbbi eset a jelző implicit jelentés-kiterjesztésének olyan módjával lenne megragadható, ami az eredményezett tartományban tartalmazná a beszédszereplők referenseit is. Vajon minden input kontextus esetén lehetséges ez? Igen, mivel a beszédszereplők referensei az interpretáció minden típusában szükségszerűen aktívak – ugyanabban az értelemben, ahogy a szakaszban leírt áthidalási mechanizmusok révén aktiválódnak korábbi jelentéselemek. „Technikailag" tehát minden referenshorgonyzási procedúra során ott vannak potenciális célpontként a beszédszereplők, valamint az ITT és a MOST referensei.

3. Az igazság pillanatának implementálása

Az előző három év elméleti alapozó munkálataira [6, 12, 15] támaszkodva a ℜeALIS implementálását egy igazságértékelést – statikus interpretációt – végző program megírásával kezdtük, amit majd ki tudunk bővíteni dinamikus interpretációt is elvégző programmá [13]. Az alapprogram hátterét ismertetjük a következő oldalakon.

Az igazságértékelés magvát egy mondat reprezentációjának a világ modelljére való ráillesztése jelenti. Új ötletként merült fel a kutatás egy pontján, hogy induljunk ki a külvilág modelljéből, ne a nyelvből.

A ℜeALIS az időt is figyelembe veszi, ezért egy „világtörténelmet" kell modellezni. Így hát az univerzum entitásai, entitáspárjai, entitáshármasai... mellett minden külvilágbeli magreláció egy időintervallumot is tartalmaz. A *kopasz* relációnak például lehet egy eleme ilyesmi: ⟨[20060406,20090711], Péter⟩; ami mellett elképzelhető egy ilyen elem is: ⟨[20131220,20140507], Péter⟩. A szomorú „sztori" megfejtése: bő három év kopaszság után Péter 2009 júliusában egy néhány évig sikeres hajbeültetésen esett át, majd élete utolsó hónapjaiban újra kopasz volt. A *havazik* pedig például egy-egy térreferenst párosít egy időintervallummal.

A ℜeALIS megközelítésének [3] döntő eleme, hogy heterogén jelentésszerkezetet tükröző relációkat nem kívánunk alkalmazni a világmodellben. A nyelv heterogén jelentésszerkezetű kifejezéseiről úgy kívánunk számot adni, hogy az interpretálói elmékbe jelentésposztulátumokat „ültetünk". A *hazamegy* interpretációját például a *megy*, *lakik*, *kívül van*, *belül van* homogén relációk vizsgálatára alapozzuk, különös tekintettel a társított időintervallumok viszonyaira.

Kicsit előreszaladva itt szeretnénk elbüszkélkedni azzal, hogy a befejezett „hazament" és a progresszív „ment haza" kifejezések értékelését el tudjuk végezni. Az előző bekezdésben felvázolt homogén relációk a befejezett változat interpretációjához elegendőek. A progresszív változat interpretációjához a cselekvő és a beszélő elmereprezentációjának vizsgálata is szükséges: az előzőnek a referenciaidőbeli szándéka, az utóbbinak a „valószínűsítő" hozzáállása szükséges. Vagyis úgy modellezzük ezáltal a Péter által kimondott „Mari éppen ment haza" mondat jelentését, hogy az akkor igaz, ha a referenciaidőt megelőzően Mari nem volt otthon, ment-mendegélt, Péter valószínűnek tartja, hogy a referenciaidő pillanatában otthon lenni vágyott, és ennek elérését Péter valószínűnek tartja vagy tartotta egy korábbi pillanatban.

Itt jegyezzük meg, hogy a jelentésposztulátumok finomhangolását az elkövetkezendő évek feladatának tartjuk, valamiféle „empirikus szemantika" útjait törve ezáltal. Első közelítésben annyi a lényeges, hogy a ℜeALIS világmodelljének egyenrangú részét képezi a külvilág relációs hálózatának és az interpretálói elmék relációs hálózatának reprezentációja, és így egyetlen összetett mintaillesztési eljárás során a külvilágbeli magrelációkat egyidejűleg vizsgálhatjuk az elmék α, σ és λ relációival.

A 2. szakaszban tárgyalt jelentésspecifikáció is e belső relációk hálózatának vizsgálatára épül. Egy-egy referens körül táguló körökben keressük a megfelelő típusú referenst; és amikor megtaláljuk, akkor a hozzávezető utat az előző bekezdésben emlegetett jelentésposztulátumként foghatjuk fel. Ha például az egyetem testületként van jelen a magrelációk körében, akkor az „egyetem után" kifejezés értelmezése azt igényli, hogy a testülettől eljussunk egy olyan referensig, amelyre igaz az az állítás, hogy

időtartam. Nyilván olyan elmereprezentációt kell ehhez feltételezni, amelyben összetársulnak a következő jelentésdarabok: a testület diákokat oktat, akiknek tanterve van, aminek teljesítése tipikusan néhány évet igényel.

A program magva tehát egy lekérdező interfész, ami a „belső felhasználóktól" homogén relációkat kér a kezdetben strukturálatlan entitáshalmazként definiált és csak időintervallumokkal felruházott külvilág benépesítésére. Az emberi entitások esetében pedig mindig rákérdez, hogy a *kopasz*, *férfi*, *ukrán*, *valamin kívül van*, *valakit szeret* típusú relációkon kívül milyen α, σ és λ relációs kapcsolatok fűzik bizonyos entitásokhoz – amelyek éppen e kapcsolatok révén válnak belső mentális entitásokká, azaz a szóban forgó személy referenseivé. Az α reláció fontos feladata még a referensek „kihorgonyzása". Ennek alapesete az észlelés modellezése: Péter megpillantása (vagy másfajta észlelése) az adott pillanatban egy addig izolált belső entitás „használatba vételét" váltja ki, vagyis kihorgonyzását Péter külvilágbeli entitására. Az igazságértékelés során éppen ezt a horgonyt használjuk a Péterről szóló állítások esetében.

Hogy ne tűnjön parttalannak az elmék belső relációs struktúrájának a kiépítése, éljünk azzal az egyszerűsítéssel, hogy csak véges sok pillanatban mutatnak eltérő információállapotot, egy adott időpillanatban pedig minden külvilágbeli reláció σ_0 reifikált változatához predikátumreferensek horgonyzódnak. Ez a hármas például benne van a σ_0 relációban egy korábbi bekezdés tartalmát felhasználva: ⟨kopasz, 20081111, Péter⟩. Vegyük észre, hogy az időmetszetek e módszere miatt volt szükséges homogén relációkra korlátozódnunk – ezzel azonban semmilyen nyelvi jelenség megragadásáról nem mondtunk le, ugyanis a nyelvi kifejezések csak egy elmében reprezentálhatóak, ott pedig jelentésposztulátumok segítségével modellezni tudjuk a heterogén jelentésszerkezetet.[3]

Továbbra is küzdve az elmereprezentáció parttalansága ellen, a világtörténelem modelljének megválasztása kiindulópontnak felveti a lehetőségét annak, hogy egy orákulum szerepű interpretáló elméjét automatikusan legeneráljuk néhány kijelölt időpillanatban. Ennek módja: a σ_0 adott pillanathoz kötődő elemeit észlelt infonnak tekintve másoljuk be az orákulum gyökérvilágába egy-egy eventualitásreferenshez társítva a σ segítségével, ahogyan azt az (1c) példában szemléltettük.

Az orákulum tehát minden külső magrelációt észlel egy-egy pillanatban, de λ relációja üres: nincsenek hiedelmei, vágyai, szándékai. Egy hús-vér ember információállapotát ez alapján úgy állíthatja elő a belső felhasználó, hogy az orákulum fejéből „átmásolt" relációs struktúrát gazdag λ szintrelációval társítja, a 2011-ben bemutatott [5] „prizmákba" helyezve az eventuális referenseket. Nyilván csak az eventualitások töredékét érdemes 0-tól különböző tudáspolaritással felruházni, így számot adva a hús-vér ember erősen részleges világismeretéről. Míg az orákulum tudja, hogy Péter otthon van, Juli mondjuk csak valószínűsíti ezt, ugyanakkor olyan vágya is van ezzel kapcso-

[3] Arra is felhívjuk az olvasó figyelmét, hogy a σ_0 reifikált relációnak az elmék belső struktúrájában szerepet játszó (hasonló szerepű: „állítás-építő") σ relációval való egyesítése után – jelölje ennek eredményét σ^* – a teljes külső és belső univerzum entitásai között három reláció marad: α, λ és σ^*. A dinamikus interpretáció céljaira később majd bevonjuk a ℜeALIS κ „kurzorfüggvényét" is, modellezendő az aktív elmezónákat.

latban, hogy Mari úgy gondolja, hogy Péter nincs otthon, sőt azt gondolja, hogy ő is éppen ezt gondolja, stb.

„Belső felhasználókat" említettünk korábban: itt elsősorban kutatócsoportunk nyelvész tagjairól van szó, de mások is kialakíthatják a külvilág modelljét, meghatározhatják, hogy hány időpillanatra hány interpretálói „elme" tartalma legyen feltöltve, és milyen nyelvi jelenségek megragadása céljából. Az elméket érdemes hasonló kulturális, enciklopédikus és logikai jellegű tudással ellátni, persze ezt is az α, σ és λ relációs kapcsolatok „nyelvén" megfogalmazva, a jelentésposztulátumokhoz hasonlóan. Ugyanakkor érdekes vizsgálatokra ad lehetőséget, ha az interperszonális tudások eltérésén túl az előbb említett tudásfajták tekintetében is elhelyezünk különbségeket, modellezni próbálva műveltség- és intelligenciabeli eltéréseket.

Olyan különbséget is elhelyezhetünk a modellezett interpretálói elmék között, hogy az egyikben tucatnyi jelentésváltozatát felvesszük az *egyetem* vagy a *háza* kifejezéseknek az (1a) mondat kapcsán tárgyaltak alapján, míg a másikban az áthidalási mechanizmusokra bízzuk a megfelelő változat elérését.

A „belső felhasználókhoz" képest olyan külső felhasználókat is számításba veszünk, akik az információval feltöltött rendszert kapják, amit igazságértékelésre használhatnak. Beírva olyan kijelentő mondatokat, mint az alábbiak a 3. példában, kapnak egy „igaz" vagy „hamis" választ, igény szerint kiválasztható megjegyzéstípusok kíséretében.

Tekintsük az alábbi (3a) mondatot! Csakis akkor lehet értékelni, ha tudjuk, hogy ki mondta, és mikor. A külső felhasználónak tehát ki kell választania az eddigiekben felvázolt rendszerben egy időpillanatot („most") és egy interpretálót („én"), akinek a szájába adja a mondatot. A gép a külvilág modellje alapján dönt az igazságérték felől. Mivel a *havazik* relációban egy-egy térreferens párosul egy időintervallummal, az a kérdés, hogy a „most" pillanatát tartalmazó valamely időintervallum egy olyan térreferenssel párosul-e, amelyben ott van az „én" tartózkodási helye.

3. példa. MONDATOK IGAZSÁGÉRTÉKELÉSRE

a. Havazik.
b. Szerintem Petya úgy tudja, hogy a kopasz ukrán férfi ismeri Petit.
c. Petya tudja, hogy Szása nős.

A (3b) mondat azt is egyértelművé teszi, hogy az igazságértékelést kérő külső felhasználónak a címzettet is meg kell jelölnie, „a kopasz ukrán férfi" ugyanis csak akkor egyértelmű horgonyzó információ, ha a beszélő egyetlen személyről gondolja, hogy kopasz is, ukrán is, férfi is, és/vagy a közlés címzettjéről is feltételezi ugyanezt az unicitási helyzetet. A gép „hibaüzenetet" küld, amennyiben akár a beszélő, akár a címzett nem pontosan egy kopasz ukrán férfit „horgonyoz le". Ez persze azt az egyszerűsítő feltételezést igényli, hogy a ℜeALIS világtörténelem-modellje gyakorlatilag inkább egy gazdagabb szituáció leírásának felel meg.

A „Petya" és a „Peti" becenevek is a beszélő és a címzett pontos megjelölését igénylik. Lukafalvi Péterről csak néhányan gondolják azt, hogy ő „Petya", és ugyanez a helyzet Mátyuska Péter „Peti"-ként való aposztrofálásával. A (3b) mondat igazságértékelését csak akkor kezdheti el a gép, ha a kiválasztott beszédpartnerek megegyez-

nek e becenevekben. Vegyük észre, hogy pontosan ugyanarról van szó, mint „a kopasz ukrán férfi" horgonyzási hatékonysága ügyében.

Megjegyzésre érdemes, hogy még az sem kizárt, hogy a beszélő hiedelmei szerint ukrán férfi – mondjuk Szása – valójában orosz. A beszélő tehát jóhiszeműen jár el, amikor Szásáról azt állítja, hogy „Petya szerint Szása ismeri Petit". Horgonyozni a beszélő belső világa alapján kell! A gép ilyen üzenettel értékel: „Az állítás igaz: Petya szerint Szása ismeri Petit; de tévedésen alapul a horgonyzás."

A (3b) mondat igazságértékelését nem a külvilág relációi alapján kell végrehajtani, hanem a „szerintem" és az „úgy tudja" kifejezések – mint valami útjelző táblák – által megjelölt helyen. A „szerintem" a beszélőnek választott interpretáló elméjének reprezentációjába „küld", amin belül egy „Petyának" tulajdonított hiedelmet kell megkeresni. A λ reláció megfelelő címkéi mentén kell bejutni a megfelelő világocskába. A mondat pontosan akkor igaz, ha ott egy eventualitásba bele van foglalva, hogy egy Szásához horgonyzott referens és egy Mátyuska Péterhez horgonyzott referens az „ismeri" predikátumreferensével társul. A gép külön kérésre jelezheti, hogy a mondat ugyan igaz az adott modális kontextusban, de anélkül (a külvilágra vetítve) hamis az, hogy „Szása ismeri Mátyuska Pétert". Ebből az is kiviláglik, hogy a gép nemcsak értékeli, de értelmezi is a mondatot – ami tulajdonképpen nem jelent többletfeladatot, hiszen az igazságértékelést csakis a horgonyzási részletek tisztázását követően lehet elvégezni.

A fenti (3c) kapcsán egy újdonságra szeretnénk kitérni (miután a horgonyzási problémakört az imént kivesézték). A mondat értékeléséhez választott pillanatban kell Petya elméjének reprezentációját átfésülni.[4] Mi van, ha egy jóval korábbi pillanatban található csak meg a Szása nős mivoltát közlő eventualitás Petya elméjében? Azt a helyzetet modellezzük, hogy Petya 10 éve hallott utoljára Szásáról – aki akkoriban közismerten nős volt.

A gépnek pontosan erről a helyzetről kell tudósítania, hiszen a mondat szorosan véve hamis (Szása 10 év alatt tízszer elválhatott, Petya tehát nem tudhatja biztosan, hogy mi az igazság); ugyanakkor releváns azért a korábbi tudás! Például ha Petya két napja értesült róla, hogy Szása nős, akkor nem lenne szerencsés úgy értékelni a helyzetet, hogy „Petya jelenleg nem tudja az igazságot". A való világban szinte mindig korábbi információ alapján kell döntéseket hoznunk. A kérdés tehát az, hogy mely állítás mennyi idő alatt tekintendő „elévültnek".

Hozzárendelhetjük ezt az elévülési időt minden predikátumhoz; elegáns választ nyújtva azokra a kérdésekre, hogy egy adott pillanatban melyik interpretáló mit tud. Annyi szükséges csupán, hogy a tudásról szóló mondatokat ne csak a „most" pillanatában értékeljük, hanem a korábbi pillanatokban is.

További mondatokat a bemeneti szintaxis ismertetését követően fogunk tárgyalni.

[4] Vegyük észre, hogy csak „Petya" horgonyzása történik a beszélő információállapota alapján, az állítást már nem ott kell értékelni, hiszen nem úgy kezdődött a mondat, hogy „szerintem..."

4. Egy lokálisan korlátlan szintaxis

A jelenlegi fázisban a program dinamikus szemantikai interpretációra még nem alkalmas, azaz nem tud egy input szöveget egy beszélői információállapotban „felépíteni”, új információállapotot kiszámítva. A puszta igazságértékeléshez elegendő egy olyan (végtelen) mondattömeg, ami éppen a modellezett nyelvi jelenségeket mutatja fel.

Az alábbi szintaxis néhány mondatklisét kínál (4a), úgy téve korlátlanná a lehetőségeket, hogy egyes klisék újabb mondatokat (S) és hasonló rekurzióra alkalmas (4b) főnévi kifejezéseket (D) kérnek. Az előző szakaszban láttuk, hogy a horgonyzás a beszélő információállapotán múlik, míg az állítást magát a külvilágból kiindulva kell értékelni, ahonnan bizonyos kifejezések hatására be kell lépni meghatározott elmék világocskáiba. Hogy mely szavak szolgálnak a horgonyzás céljaira, azt pontosan körül tudjuk határolni a megadott szintaxisban: a D-ből levezetettek.

4. példa. A SZINTAKTIKAI BEMENET

a. S lehetőségei:
   1. D szerint / énszerintem / teszerinted S
   2.a-e D tudjaT / úgy tudja / arra vágyik / valószínűsíti / rájön, hogy S
   3.a havazik / fagy
   3.b D kopaszT / ukránT / magyarT / nősT
   3.c D csinosT
   4.a D megyT / röpülT / úszikT
   4.b D ismeriT / szeretiT D-t
   4.c D hazamegyT / beröpülT
   5.a D elássaT D-t
   5.b D elásatja D-t D-vel D-be
   6.a D belefog D-be
   6.b D befejezi / félbehagyja / folytatja D-t
   7.a nem áll fenn S
   7.b. S és / vagy S
b. D lehetőségei:
   1.a egy/a/minden A1 A2 fiú/ház
   1.b egy/a/minden A1 A2 háza/lánya D-nek
   1.c egy/a/minden A1 A2 könyv D-ről
   2. Péter / Peti / Petya / Mari / Juli
   3. az a A1 A2 fiú/ház
c. T lehetőségei: bef-jelen, bef-múlt, foly-jelen, foly-múlt, felt-jelen, felsz
d. A1 lehetőségei: ∅, magas, kopasz
e. A2 lehetőségei: ∅, magyar, ukrán

A következő példák kapcsán néhány nyelvi jelenség kezelését fel tudjuk vázolni. Ezek köre a rendszer bővítésével egyre bővíthető.

5. példa. MONDATOK IGAZSÁGÉRTÉKELÉSRE

a. Befejeztem a házát a lányának Marinak.
b (Szerintem) a magas ukrán lány csinos.
c. Mari arra vágyik, hogy havazzon és fagyjon.
d. Rájöttem, hogy az a szőke lány ismeri azt a magas férfit.
e. Hazudsz!
f. Lelőtted a poént!

Az (5a) mondat furcsa változatban kínálja a birtokos szerkezet kezelését. Ennek előnye az, hogy így iterálni tudjuk a birtokos szerkezeteket a (4b.1b) alapján.

A „befejez"/„nekifog" igetípusról a 2. szakaszban sok szó esett. A külvilágban közvetlenül nincs nekik megfelelő reláció. A gépnek javaslatot kell tennie az aktuális jelentésposztulátumokra nézve; azaz a példában a beszélői elmében a „ház"-hoz asszociálódó cselekvéseket kell keresnie. Az adott információállapoton múlik – ezt tudjuk ezzel modellezni –, hogy a ház felépítéséről, bevakolásáról, kitakarításáról vagy egyébről van-e szó.

Az (5b) kapcsán (először „szerintem nélkül") arról szeretnénk értekezni, hogy a „csinos" predikátumnak szintén nem jogos egy külvilágbeli relációt megfeleltetni. Jelentésposztulátum helyett – vagy gyanánt – ezúttal a következőt javasoljuk: a gép pásztázza végig valamennyi interpretálói elmét, és értékeljen a „közösség" véleménye alapján. Mintha azt mondanánk: csinos az, akit az emberek 66%-a annak tart. Számít azért a beszélőnek választott interpretáló véleménye is: ha ő nem gondolja csinosnak az érintettet, akkor így fogalmazna: „csinosnak mondják".A „szerintem" beillesztésével alkotott mondatot viszont elég kizárólag a beszélői elme tartalma alapján igazságértékelni.

Az (5c) interpretációjának kulcsa: Mari megfelelően megcímkézett vágyvilágocskájában megkeresni a „havazik és fagy" eventualitást. A ℜeALIS rendszerében egy akkommodációs lépés állítja elő a „havazik" eventualitásból és a „fagy" eventualitásból a konjunkcióhoz tartozót. A gép jelezni tudja e kettő meglétét akkor is, ha a harmadik nincs meg, és utalni tud arra, hogy Marin múlik, hogy hajlandó-e a logikai lépés megtételére. Akinek ez a gondolatmenet erőltetett, az gondoljon bele, hogy bonyolultabb következtetési sémák alkalmazása már igenis személyfüggő. Ezt kívánjuk megragadni.

Az (5d) a „rájöttem" révén olyan vizsgálatát követeli meg a beszélői információállapotnak, ami több időpillanatra is kiterjed. A mellékmondatban megfogalmazott e eventualitást kell megtalálni egy korábbi időpillanatban nem pozitív polaritásértékkel, egy későbbiben pedig pozitívval. A ℜeALIS implementációjában ez az intenzionális helyzet az elmék modellbe épített reprezentálása folytán semmiben nem különbözik egy extenzionális kvantálás vizsgálatától.

A mutató névmást a gép úgy ragadja meg, hogy egy entitáscsoport kijelölését kéri a külső felhasználótól. Sikeres a horgonyzás, ha abban pontosan egy szőke lány van, és pontosan egy magas férfi. Vagy ha legalább a beszélő így gondolja.

Programunk úgy is tudja hasznosítani a megcímkézett elmevilágocskákba ültetett információt, hogy a „beszélő" S mondatának néhány, a címzett szájába adható pragmatikai értékelését kínálja igazságértékelésre. A hazugság (5e) úgy mutatható ki, hogy

vizsgálatnak vetjük alá a beszélő hiedelemvilágocskájának a tartalmát: vajon milyen polaritási előjellel tárolódik ott az S-nek megfelelő eventualitás. Csupán világocskacímkék összevetését kívánja meg az (5f)-ben közölt bonyolult pragmatikai helyzet igazságértékelése is: azt kell megvizsgálni, hogy az értékelt S kijelentés negatív előjellel szerepel-e az „arra vágyik, hogy megtudja” címkéjű világocskában az adott pillanatban, ellentétben egy későbbi pillanattal.

5. Záró megjegyzések

A jelenlegi fázisban lévő program jó alapot jelent a dinamikus szemantikai interpretációval való kiegészítésre: azaz alkalmas lesz majd egy beszélői információállapotban „felépíteni” egy szöveget, új információállapotot kiszámítva. Ugyanis ennek kulcsa is a ℜeALIS [2–3] gazdag kül- és belvilág-modelljéhez való hozzáhorgonyzása a mondatok preszuppozíciós zónájának, ami az ott található információ igazságértékelésén alapul (szintaktikai kutatásaink eredményeit is alkalmazva[2, 6]).

Említünk egy érdekes alkalmazási területet. Nyomozási vagy bírósági folyamatokat kívánunk modellezni, bizonyos döntő pillanatokat kiragadva, melyekben regisztráljuk az (akár párbeszédeket is tartalmazó) dokumentáció alapján, hogy melyik szereplő mit állít, és vélhetőleg mit tud a külvilágbeli történésekről, illetve mások feltételezéseiről és vágyairól, szándékairól. Erre vonatkozóan lehet majd állításokat értékeltetni, illetve a külvilági „igazságot” is felépíteni az e cikkben leírt orákulumképzési eljárás inverze segítségével, továbbá felépíteni az egyes szereplők információállapotainak reprezentációját.

Hivatkozások

1. Alberti G.: Mi fán terem a „konkordiális nyelvtan”? In: Büky L., Maleczki M. (szerk.): A mai magyar nyelv leírásának újabb módszerei IV. SZTE, Szeged (2000) 9–41
2. Alberti G.: ℜeALIS, avagy a szintaxis dekompozíciója. Általános Nyelvészeti Tanulmányok XXIII. (szerk. Bartos H.) (2011) 51–98
3. Alberti G.: ℜeALIS. Interpretálók a világban, világok az interpretálóban. Akadámiai Kiadó, Budapest (2011)
4. Alberti, G.: Where are Possible Worlds? II. Pegs, DRSs, Worldlets and Reification. In: Alberti G., Kleiber J., Farkas J. (szerk.): Vonzásban és változásban. PTE Nyelvtudományi Doktori Iskola (2012)
5. Alberti G.: Az intenzionalitás számítógépes nyelvészeti kezelése – avagy a ℜeALIS λ szintfüggvénye. In: MSzNy 2011. SzTE Informatikai Tanszékcsoport, Szeged (2011) 263–275
6. Alberti G., Kilián I.: Vonzatkeretlisták helyett polaritásos hatásláncсaládok – avagy a ℜeALIS σ függvénye. In: MSZNY 2010. SzTE Informatikai Tanszékcsoport, Szeged (2010) 113–126
7. Alberti, G., Kleiber J.: The Grammar of ℜeALIS and the Implementation of its Dynamic Interpretation. Informatica, Vol. 34, No. 2 (2010) 103–110

8. Alberti, G., Kleiber, J.: Where are Possible Worlds? (Arguments for ℜeALIS). Acta Linguistica Hungarica, Vol. 59, No. 1-2 (ed. Katalin É. Kiss) (2012) 3-26
9. Asher, N., Lascarides, A.: Logics of Conversation. Cambridge Univ. Press (2003)
10. Dowty, D. R., Wall, R. E., Peters, S.: Introduction to Montague Semantics. D. Reidel Publishing Company, Dordrecht (1981)
11. Kamp, H., van Genabith, J., Reyle, U.: Discourse Representation Theory. In: Gabbay, D., Guenthner, F. (eds.): Handbook of Philosophical Logic, Vol. 15. Springer-Verlag, Berlin (2011) 125–394
12. Károly M.: Interpretáció és modalitás – avagy a ℜeALIS λ-függvényének implementációja felé. In: Tanács A., Vincze V. (szerk.): VIII. Magyar Számítógépes Nyelvészeti Konferencia, MSzNy 2011. SzTE Informatikai Tanszékcsoport, Szeged (2011) 284–296
13. Károly M.: A ℜeALIS statikus interpretációjának kísérleti implementációja. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 318–323
14. Kiefer, F.: Jelentéselmélet. Corvina, Budapest (2000)
15. Kilián I.: Tárgymodell változatok a ℜeALIS nyelvi elemzéshez. Tanács A., Vincze V. (szerk.): VIII. Magyar Számítógépes Nyelvészeti Konferencia, MSzNy 2011. SzTE Informatikai Tanszékcsoport, Szeged (2011) 276–283
16. Landman, F:. Towards a Theory of Information. Foris, Dordrecht (1986)
17. Laczkó T.: Az ige argumentumszerkezetét megőrző főnévképzés. Kiefer F. (szerk.): Strukturális magyar nyelvtan. I. Morfológia. Akadémiai Kiadó, Budapest (2000) 293–452
18. Pustejovsky, J.: The Generative Lexicon. MIT Press, Cambridge, Mass. & London (1995)

VII. Információkinyerés és -visszakeresés

Kulcsszókinyerés alapú dokumentumklaszterezés

Berend Gábor[1], Farkas Richárd[1], Vincze Veronika[2],
Zsibrita János[1], Jelasity Márk[2]

[1]Szegedi Tudományegyetem, TTIK, Informatikai Tanszékcsoport
Szeged, Árpád tér 2., e-mail:{berendg, rfarkas, zsibrita}@inf.u-szeged.hu
[2]Magyar Tudományos Akadémia, Mesterséges Intelligencia Kutatócsoport
Szeged, Tisza Lajos körút 103., e-mail:{vinczev, jelasity}@inf.u-szeged.hu

Kivonat A szöveges dokumentumok lényegi mondanivalóját tömören összegezni képes kifejezések kitüntetett fontossággal bírnak: számos nyelvtechnológiai alkalmazás profitálhat ismeretükből a katalogizáló és kivonatoló rendszerekben történő felhasználásuktól kezdve egészen az információ-visszakereső alkalmazásokig. Cikkünkben automatikusan meghatározott kulcsszavak minőségét alternatív módon, egy dokumentumklaszterező alkalmazásban való felhasználásuk kapcsán vizsgáltuk. A munkánk során felhasznált dokumentumokat a *Magyar Számítógépes Nyelvészeti Konferencia (MSzNy)* megjelent konferenciakötetеinek cikkei képezték. A cikkekből történő csoportképzést összehasonlítottuk a cikkekben előforduló n-gramok, valamint gépi tanulás útján meghatározott kulcsszavak alapján is. Eredményeink tükrében kijelenthető, hogy a kulcsszavak hasznosak a dokumentumklaszterezés feladatának megsegítésében is. A cikkek automatikus kulcsszavai alapján értelmezett hasonlósági gráf vizualizálása és klaszterezése során tapasztaltak alapján megfigyelhető volt továbbá a nyelvtechnológia egyes részterületeinek elkülönülése, időbeli fontosságuk változása, amely alapján az automatikus kulcsszavak – alkalmazásoldali szempontból – megfelelő minőségére következtethetünk.

Kulcsszavak: automatikus kulcsszókinyerés, dokumentumklaszterezés

1. Bevezetés

A dokumentumokhoz – automatikusan avagy manuálisan – rendelt kulcsszavak azon túl, hogy egy tömör összefoglalójaként értelmezhetők az egyes dokumentumoknak – és ezáltal alkalmassá válnak azok visszakeresésének vagy osztályozásának megkönnyítésére –, fontos eszközei lehetnek a dokumentumok közötti hasonlóságok meghatározásának. Jelen cikkben azt a kérdést vizsgáljuk, hogy a dokumentumok között definiált hasonlósági reláció modellezésére alkalmasabb-e az egyes dokumentumok kulcsszavaira támaszkodni, mint a hagyományos vektortérmodellre (ahol a dokumentumokat a bennük előforduló összes n-grammal jellemezzük).

2. Kapcsolódó munkák

Az elmúlt években számos tudományos eredmény látott napvilágot hazai és nemzetközi szinten egyaránt a dokumentumok lényegét leírni hivatott kifejezéshalmazok automatikus meghatározását végző rendszerekre nézve. Ezen munkák jellemzően angol nyelvű tudományos publikációk kulcsszavainak automatikus meghatározását tűzték ki célul (pl. [1],[2] és [3]), azonban akadnak kivételek is, amelyek más doménű dokumentumok kulcsszavazására vállalkoztak (pl. [4], [5] és [6]). Mindamellett, hogy az angol nyelvű tudományos publikációkból történő kulcsszókinyerésnek tehát igen bő irodalma áll rendelkezésre, magyar nyelvű politika- és neveléstudományi témában íródott tudományos publikációk kulcsszavainak gépi tanuláson alapuló meghatározására is született már kísérlet [7].

A korábbi munkák hatékonyságának objektív megítélésének komoly gátat szab az a tény, hogy a kulcsszavak minőségének emberi elbírálása meglehetősen szubjektív, valamint az automatikus (szigorú sztringegyezésen alapuló) kiértékelésük szintén nehézségekbe ütközik az azonos (szinonim) vagy közel azonos (hipo- vagy hipernim) jelentésű kifejezések megjelenési formáinak sokszínűsége kapcsán. Jelen munka egyik célja egy alternatív kiértékelési lehetőség definiálása a kulcsszavak minőségének megítélésére, amely során a kulcsszavazás hatékonysága azon keresztül kerül lemérésre, hogy milyen mértékben sikerül egy korpuszt alkotó dokumentumokat elkülöníteni egymástól, csupán a hozzájuk tartozó legmegfelelőbbnek ítélt kulcsszavak ismeretének fényében.

A tudományos trendek természetesnyelv-feldolgozási eszközökkel történő kutatásának témájában szintén születtek már korábbi munkák. Ezek közül egy [8], ahol kulcsszavakhoz hasonló kifejezések előfordulásainak időbeli változását nyomon követve határozták meg a különböző tudományos résztémakörök relatív fontosságának változását.

3. Módszertan

A következő alfejezetek azt mutatják be, hogy az MSzNy-cikkarchívum egészének automatikus kulcsszavazása miként zajlott, majd ezt követően az egyes cikkekhez rendelt kulcsszavak alapján hogyan lettek azonosítva az egyes számítógépes nyelvészeti részterületek.

3.1. Automatikus kulcsszavazás

Mivel a cikkek szerzői csupán az esetek elenyésző hányadában látják el írásukat az azt jellemző kulcsszavakkal, ezért ahhoz, hogy a dokumentumok klaszterezése az őket legjobban leíró kulcskifejezések alapján is megtörténhessen, szükség volt egy olyan modell építésére, amely képes a kulcsszavak cikkek szövegéből történő automatikus kinyerésére. A feladat megoldása alapvetően a [7] által ismertetett módszert követte. A kulcsszavak meghatározására először a dokumentumból kigyűjtöttük a lehetséges kulcsszójelölteket, majd felügyelt tanulási módszerekkel azokat fontossági sorrendbe rendeztük. Jelen esetben a rangsorolás egy bináris

valószínűségi osztályozó a posteriori valószínűségein alapul, ahol a bináris osztályozót arra tanítjuk, hogy egy kulcsszójelölt szerepelt-e a dokumentum szerzője által a szóban forgó dokumentumhoz rendelt kulcsszavak között vagy sem. Ez a bináris tanuló a [7] jellemzőkészletéhez hasonló módon pozicionális, ortografikus és morfológiai jegyeik alapján reprezentálta a kulcsszójelölteket, az osztályozásukhoz pedig maximum entrópia modellt használtunk. A morfológiai elemzés elvégzésére a [9] modelljeit használtuk föl.

3.2. Dokumentumok hasonlóságának mértéke

Két dokumentum hasonlóságának mérésére több módszert is vizsgáltunk. Egyrészt ez a hasonlóság alapulhat az előző fejezetben bemutatott automatikus kulcsszavakon, vagy a dokumentum n-gramjain
($1 \leq n \leq 2$). Mindkét megközelítésre igaz, hogy egy dokumentumot a 10 „legjellemzőbb" kifejezésével írtunk le. Az n-gramok esetén a rangsoroló mérték a hagyományos tf-idf mutató volt, míg a kulcsszavakra támaszkodó reprezentáció esetében a bináris osztályozónk a posteriori valószínűsége volt mindez.

Két dokumentum esetén akkor beszélünk pozitív hasonlóságról, ha azok legalább egy közös „jellemző" kifejezéssel rendelkeznek. Két dokumentum kulcsszavaiból álló halmaz metszetének értékelésére több stratégiát is alkalmaztunk: egyes esetekben a mindkét halmazban megtalálható kifejezések fontosságértékének *maximumai*, *minimumai*, *átlagai*, *szorzatai*, illetve *harmonikus közepei* lettek véve, majd a két dokumentum globális hasonlóságának meghatározásához ezek az értékek összegezve lettek átfedésben álló kifejezéseik fölött. Két további megközelítés az átfedő kifejezések fontosságát nem, csupán azok számosságát vette figyelembe: ezek a *Dice*- és *Jaccard*-együtthatókon alapuló módszerek voltak.

3.3. Hasonlósági gráf alapú klaszterezés

Végső célunk egy dokumentumhalmaz klaszterezése, melyhez a csúcsaiban dokumentumokat reprezentáló (irányítatlan) hasonlósági gráfot építünk fel, a gráfban szereplő éleket pedig oly módon súlyoztuk, hogy azok értékei az előző fejezetben bemutatott páronkénti dokumentumhasonlóság-értékek voltak. Két dokumentumnak megfeleltethető a, b csúcs között csak akkor vezet él a gráfban, ha kulcsszavaik metszete nem üres, valamint az átfedés mértékét számszerűsítő súlyozás alapján b az a dokumentumhoz leghasonlóbb 3 dokumentum között szerepel vagy fordítva (a szerepel a b-vel legnagyobb hasonlóságot mutató 3 dokumentum között). A klaszterezést (particionálást) ezen a gráfon hajtjuk végre.

Egy adott gráfparticionálást jellemző modularitás [10] kiszámításával egy jósági értéket rendelhetünk a felbontás minőségére nézve, mely figyelembe veszi a gráf topológiájából adódóan az egyes csúcspárok között elvárható élek számát, valamint egy tényleges felbontás során az egyes csoportokon belül vezető élek tapasztalt számát:

$$Q = \frac{1}{2m} \sum_{ij} \Big(A_{ij} - \frac{k_i k_j}{2m} \Big) \delta(C_i, C_j), \tag{1}$$

amelyben az összegzés minden *lehetséges* élre (minden i és j csúcsra) vonatkozik, és ahol az A_{ij} a particionálandó gráf szomszédsági mátrixának egy eleme, m a gráfban található élek száma, az összegzésben található hányados pedig az i és j csúcsok összeköttetésének – k_i és k_j fokszámértékekre támaszkodva számított – várható értéke, a δ függvény pedig az ún. Kronecker-delta, mely akkor veszi fel az 1 értéket, ha az i és a j csúcsok megegyező klaszterbe soroltak, egyébként 0.

Egy gráf olyan felbontásának meghatározása, amely erre a mutatóra tekint maximalizálandó célfüggvénye alapjául, erősen $\mathcal{NP}$-teljes [11]. Több közelítő eljárás látott már azonban napvilágot a probléma minél hatékonyabb, gyors megoldására, melyek között találunk szimulált hűtéstől kezdődően spektrálmódszereken át mohó megközelítéseket alkalmazókat is.

A spektrálmódszereken alapuló eljárások hátránya a megfelelő skálázódásuk hiánya, noha az alkalmazásukkal elért eredmények gyakorta felülmúlják a más megközelítésekkel kapottakat. A [12] által javasolt mohó optimalizáló stratégia kifejezetten nagy gráfokon is működőképesnek bizonyult, így az általuk javasolt eljárást valósítottuk meg a dokumentumhasonlósági gráf particionálására. Jóllehet a kísérleteink során megkonstruált gráfok csúcsainak számai mindössze százas nagyságrendben mozogtak, abból kifolyólag, hogy a későbbiekben nagyságrendekkel nagyobb dokumentumkollekciókon is használható legyen az algoritmusunk, ezért fontosnak éreztük a particionálást elvégző eljárásnak olyat választani, amely kedvező számítási bonyolultsággal rendelkezik.

A [12] szerzői által javasolt megközelítés egy alulról-felfele építkező klaszterező eljárás, mely kezdetén minden csúcsot egy külön klaszterbe sorol, majd a további lépések alkalmával a csúcsok meglátogatása során azokat a lokálisan legjobb modularitásnövekményt eredményező közösséghez sorolják (esetleg egyikhez sem). Egy i csúcs C közösségbe történő mozgatása során kettős hatás figyelhető meg: egyrészt növeli a globális modularitás értékét azon élei által, amelyek immáron a C közösségbeli szomszédjaival való összeköttetést biztosítják, másrészről viszont a modularitás bizonyos mértékű csökkenése is megfigyelhető lesz azon élei kapcsán, amelyek a korábbi közösségének tagjaival való összeköttetésért voltak felelősek. Egy i csúcs C közösségbe történő átmozgatásának hatása a következők szerint összegezhető:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right], \quad (2)$$

ahol $\sum_{in}$ és $\sum_{tot}$ értékek rendre a C közösségen belül, illetve a C közösséget érintő élek súlyainak összege, k_i és $k_{i,in}$ pedig rendre az i csúcsot tartalmazó, illetve az i csúcsot a C közösséggel összekötő élek súlyainak összege, m pedig a particionálandó gráfban található élek összsúlya. Miután minden csúcs besorolást nyert az egyes közösségekbe, az algoritmus a kialakult közösségeket összevonva, és azokat egy csúcsként kezelve megismétli az előző eljárást. Az előzőekben ismertetett eljárás gyorsaságán túl egy további előnye, hogy a kialakuló közösségek száma a particionálandó gráf topológiája alapján kerül meghatározásra, a meg-

1. táblázat. Az MSzNy legnépszerűbb témáinak eloszlása 2003-2013 között.

	2003	2004	2005	2006	2007	2009	2010	2011	2013	Összesen	Arány
cikkek száma	59	46	52	49	32	45	46	40	42	411	
morfológia	6	6	9	2	3	3	4	7	8	48	11,68 %
beszédfelismerés	5	5	5	4	5	7	6	4	2	43	10,46 %
pszichológia	5	7	5	10	6	5	0	3	2	43	10,46 %
szemantika	3	3	3	6	3	4	7	7	6	42	10,22 %
lexikográfia	7	4	6	2	0	4	6	4	5	38	9,25 %
szintaxis	5	4	7	2	5	2	5	3	3	36	8,76 %
korpusz	4	4	5	3	3	3	3	4	7	35	8,52 %
információkinyerés	2	4	2	3	1	7	10	1	5	35	8,52%
fordítás	6	7	3	4	1	4	1	4	1	31	7,54%
ontológia	1	1	4	9	0	2	1	1	0	19	4,62%

határozni kívánt csoportok számát egyéb eljárásokkal (pl. k-közép klaszterezés) szemben nem tekinti előre ismertnek.

4. Az MSzNy korpusz

Jelen munkában az *MSzNy* eddig megjelent konferenciaköteteinek cikkeinek klasztereződését vizsgáljuk meg. Az MSzNy-cikkeknél lehetőség van a szerzőknek kulcsszavakat megadni a cikkükhöz, amely lehetőséggel mindössze 45 esetben éltek a szerzők. Az előző fejezetben bemutatott felügyelt tanulási modellt ezen a 45 cikken tanítottuk.

A konferenciasorozat 2003-ban indult, és 2008 és 2012 kivételével minden évben megrendezésre került, így összesen kilenc év alatt megjelent 411 darab cikk képezte vizsgálódásaink alapját.Ahhoz, hogy a korpuszban megjelenő fő témakörök felügyelet nélküli detektálásának eredménye számszerűsíthető legyen, elvégeztük a korpuszba tartozó cikkek egy referenciabesorolását. Az emberi erővel történő témabesorolás alkalmával minden cikkhez az arra leginkább jellemző témakategóriák lettek meghatározva, mint például *morfológia*, *lexikográfia* stb. Arra törekedtünk, hogy a témakategóriák a számítógépes nyelvészet különféle részterületeit reprezentálják, így azok cikkekhez történő hozzárendelése felfogható legyen a dokumentumok egy osztályozásának.

Az MSzNy-cikkek kézi osztályozása és tematizálása lehetővé teszi azt is, hogy megvizsgáljuk, milyen trendek uralkodtak az utóbbi években a magyarországi számítógépes nyelvészet területén. Az 1. táblázat a tíz leggyakoribb tématerülethez társítható cikkek időbeli mennyiségi eloszlását mutatja. A táblázatból kiolvasható, hogy az összesítésben tíz leggyakoribbnak mutatkozó téma az összes, humán annotáció segítségével detektált témakör hozzávetőlegesen 90%-át fedi le. A táblázatból kiderül továbbá az is, hogy a megjelent cikkek számának tekintetében a legnépszerűbb téma a morfológia volt, valamint az is, hogy szintén számos cikk született a beszédfelismerés, illetve a pszichológiai szövegfeldolgozás témaköreiben.

Érdekes azt is megfigyelni, hogy az évek során hogyan alakult a különféle témák eloszlása. A morfológia a konferenciasorozat kezdetekor, illetőleg az utóbb években tölt be különösen előkelő pozíciót. A beszédfelismerés 2009 környékén volt népszerű téma a konferencián, a fordítás elsődlegesen 2003-2004 környékén, azaz a kezdetekben foglalt el dobogós helyet, a szemantika és a korpusznyelvészet előretörése viszont az utóbbi néhány évben figyelhető meg. Az információkinyerés különösen a 2009-2010-es években virágzott, legalábbis az MSzNy-es mutatók alapján. Kiugróan jó évnek bizonyult a 2006-os a pszichológiai szövegfeldolgozás és az ontológia számára. A táblázatban már nem szereplő tématerületek közül kettőt említünk meg: a 2007-es év különösen sok beszédszintézissel foglalkozó cikket hozott, illetőleg 2010 óta az információ-visszakeresés is egyre népszerűbb, azonban e témák az összesített helyezésük alapján nem kerültek a legjobb tízbe.

Az előző megfigyeléseket természetesen árnyalja annak ismerete, hogy csupán 9 kiadványon alapulnak, továbbá, hogy az MSzNy-en az egyes témákban évenként megjelenő cikkek számára kis elemszámú mintaként tekinthetünk csupán, melyek érzékenyek lehetnek a témák relatív népszerűségén kívüli egyéb tényezőkre is, ami azt eredményezi, hogy a minták statisztikai mutatói könnyedén módosulni képesek. Egy ilyen, a trendek megfigyelését megzavarni képes jelenség lehet például egy adott témájú projekt lezárulta, és az ezzel kapcsolatos disszeminációs tevékenységek megjelenése a konferencián, mely önmagában túlreprezentálttá képes tenni időszakosan egyes területeket.

A gépi feldolgozhatóság és a kiértékelés szempontjából azonban nem bizonyult minden cikk egyformán használhatónak, így az MSzNy archívumában található 411 cikk közül nem mind került felhasználásra a továbbiakban. Egyes cikkek idegen nyelven álltak csupán rendelkezésünkre, esetleg a dokumentumból történő szöveg kinyerése nem volt lehetséges az általunk használt eszközökkel, avagy duplikátumokkal volt dolgunk. Az előző okok miatt a hasonlósági gráfot így mindösszesen 394 dokumentum alkotta.

A kézi címkézés során egy dokumentum több kategóriamegjelölést is kaphatott, amennyiben az több számítógépes nyelvészeti részterületet is érintett. Az emberi osztályozás során bevezetésre került 31 témamegjelölés közül némelyek mindössze egy-egy ízben, akkor is csupán egy másik témamegjelöléssel karöltve lett fölhasználva, így fontosságuk igencsak megkérdőjelezhető volt. Az ilyen kevéssé fajsúlyosnak mondható témával rendelkező cikkeket – valamint az összes többi olyat is, ahol egy dokumentum témája nem volt egyértelműen meghatározott az emberi jelölés által – nem vettük figyelembe a kiértékelés során, vagyis amikor az automatizált kategorizálás átfedését vizsgáltuk az emberi osztályzásával. Ezen döntés meghozatalának hátterében az a megfontolás állt, hogy az ilyen cikkek esetében még az emberi többlettudás sem volt elegendő az egyértelmű témabesorolás meghozatalához, az általunk javasolt eljárás pedig éppen ilyen egyértelmű besorolásokat tesz.

Az előzőekkel összefüggésben 46 darab automatikus kulcsszóval egyébként ellátott – és ezáltal a hasonlósági gráfban is szerepeltetett – dokumentum nem képezte részét a korpusz cikkeinek közösségkeresés által meghatározott automatikus témabesorolásának kiértékelésében. Az eredetileg bevezetett 31 témakörből

2. táblázat. Az automatikus témamegjelölés során felhasznált cikkek témáinak eloszlása.

Téma	Mennyiség	Arány
pszichológia	40	14,04%
beszédfelismerés	38	13,33%
morfológia	32	11,23%
szemantika	32	11,23%
információkinyerés	30	10,53%
fordítás	27	9,47%
lexikográfia	25	8,77%
szintaxis	24	8,42%
korpusz	20	7,02%
ontológia	17	5,96%

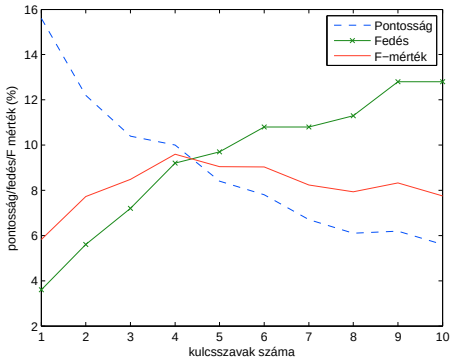
4 csupán más témák mellett kapott marginális szerepet, így a korpusz emberi kategorizálásra támaszkodó kiértékelésében is részt vevő dokumentumainak száma 337 volt, melyek 27 különböző egyedi kategóriába voltak sorolva. A 2. táblázatból kiolvasható, hogy a több kategóriába sorolt cikkek eltávolítását követően az egyes témamegjelölések hány alkalommal fordultak elő a kiértékeléshez használt adatbázisban. Megfigyelhető többek között az, hogy a megszűrt adatbázisban a leggyakoribb témának ezek után már a *pszichológia* mutatkozott, amit az okozott, hogy azon túl, hogy eredendően is viszonylag sok cikk lett hozzárendelve ehhez a kategóriához, ezek a témamegjelölések néhány kivételes esettől eltekintve teljesen egyértelműek is voltak, azaz esetükben az annotálás nem eredményezte további témák hozzárendelését a cikkekhez. Éles kontrasztot képez az előbbi témával a *morfológia* témaköre, amely előfordulásai harmadában valamely más témával együtt került megjelölésre.

5. Eredmények

Elsőként a kulcsszavazó modell hatékonyságát teszteltük, amikor is a 45 szerzői kulcsszóval ellátott dokumentum automatikusan kinyert kulcsszavainak minőségét ellenőriztük le 45-szörös keresztvalidációt alkalmazva. Egy kulcsszó elfogadása kizárólag abban az esetben történt meg, ha a normalizált alakra hozott kinyert kulcsszó tökéletes egyezést mutatott az adott cikkhez tartozó, és szintén normalizált alakban tárolt etalon szerzői kulcsszavak valamelyikével.

Megjegyzendő, hogy a 45 dokumentumhoz rendelt közel 200 kulcsszó közül mindössze 51,8% szerepelt ténylegesen is azokban a dokumentumokban, amelyekhez hozzá lettek rendelve, így a fedés értékének ez a lehető legmagasabb értéke az általunk használt kiértékelés mellett. Úgy gondoljuk azonban, hogy az eredmények ezen ténnyel való korrekciója után is a kapott számszerű eredményességi mutatók jóval elmaradnak attól a hasznosságtól, amellyel az automatikusan meghatározott kulcsszavak rendelkeznek. Mindezt arra alapozzuk, hogy a kifejezések egyezésének normalizált alakjaik szigorú sztringegyezésen alapuló vizsgálata sok szemantikai értelemben elfogadható kulcsszót álpozitív osztályba

sorolt: ilyenek voltak, amikor specializáló kifejezések nem kerültek elfogadásra a szigorú kiértékelés miatt (pl. a *felügyelt gépi tanulás* kifejezés a *gépi tanulás* ellenében), vagy amikor az elvárt és kinyert kulcsszavak jelentésükben egymással rokoníthatók voltak ugyan (adott esetben meg is egyeztek), ellenben írásmódjuk nem volt teljesen egyező (pl. a *morfológiai analízis* és *morfológiai elemzés* kifejezések).



1. ábra. A kulcsszavazó modell eredményessége a legvalószínűbbnek mondott $1 \leq k \leq 10$ kulcsszó tekintetében.

A továbbiakban már nem a kulcsszavak közvetlen minőségét, hanem használati értéküket vizsgáltuk egy dokumentumklaszterező felállásban, ahol a korpuszban szerepet kapó témákat kívántuk automatikusan meghatározni a dokumentumok szövege, illetve az abból kinyert kulcsszavak segítségével.

A cikkek által megkonstruált hasonlósági gráf particionálásának, valamint a cikkek ebből adódó automatikus témabesorolásának jóságát több mutatóval is jellemeztük. Egyrészt a közösségképzés végső minőségét számszerűsítő modularitási mutatóra támaszkodtunk. A dokumentumok particionálásának ezen mutatója csupán azt az aspektusát világítja meg az eljárásnak, hogy a hasonlósági gráfot mennyire sikerült az eredeti élstruktúrája függvényében megfelelő módon részgráfokra bontani. A megfelelőség foka azzal arányos, hogy az azonos közösségbe tartozó csúcsok között menő élek száma (vagy esetünkben azok súlyainak összege) minél nagyobb eltérést mutasson attól, mint amennyi él már csak a véletlennek is betudható lenne az egyes csúcsok fokszámai alapján.

A hasonlósági gráf magas modularitással történő felbontása azonban nem vonja feltétlenül maga után a meghatározott részkorpuszok szemantikus koherenciáját, ahogy ez a 3., valamint a 4. táblázatok kapcsán is észrevehető. Amennyiben ugyanis a csoportképződésért felelős élek olyan kulcsszavaknak köszönhetők, amelyek szemantikailag nem vagy csupán kevéssé köthetők egymáshoz, úgy kialakítható a gráf modularitás tekintetében kielégítő particionálása

3. táblázat. Automatikus kulcsszavakra nem támaszkodóan épített hasonlósági gráf particionálásának eredményei.

	Közösségek száma	Modularitás	Pontosság	V_1
Max	8	0,254	0,154	0,131
Min	9	0,372	0,160	0,127
Átlag	7	0,330	0,151	0,118
Szorzat	5	0,510	0,122	0,081
Harmonikus közép	9	0,336	0,175	0,142
Dice	2	0,071	0,113	0,025
Jaccard	2	0,072	0,113	0,025

olyan módon, hogy mindeközben a kialakult közösségek egymással rokonságba nem hozható elemekből állnak.

Éppen ezért szükségesnek éreztük további mutatók alkalmazását is a dokumentumok automatikus közösségekhez való társításának és az emberi erővel történő tematizálásuk átfedésének számszerűsítésére, ami érdekében több mutatót is alkalmaztunk. Az automatikus klasztereket leképeztük a kézzel jelölt különböző témaosztályokra, mely során mohó módon a még szóba jövő, legtöbb helyes besorolást eredményező klasztert rendeltük egy-egy etalon témaosztályhoz, amellyel egy injekciót határoztunk meg a közösségek és a témabesorolások között.

A kialakult csoportok szemantikus kohéziójának mérésére az információelméleti alapokon nyugvó V_1-mértékkel [13] is jellemeztük a kialakított közösségeket, amely a felügyelt tanulásból ismert F-mértékhez hasonlóan két érték harmonikus közepeként áll elő; a pontossággal és a fedéssel ellentétben itt a *homogenitás* és *teljesség* értékeket szokás definiálni. A homogenitás feltételes entrópiát használva számszerűsíti, hogy az egyes $c \in C$ közösségek mennyire diverzek a kézzel jelölt $k \in K$ témákhoz képest a

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{3}$$

képlet segítségével. A teljesség számítása analóg módon a

$$t = 1 - \frac{H(K|C)}{H(K)} \tag{4}$$

képlet alapján történik, a különbség mindössze annyi, hogy ennek esetében az etalon kategóriák diverzitása kerül számszerűsítésre a feltárt közösségek fényében. Egy tökéletes klaszterezés esetében az összes egy etalon témacsoportba tartozó elemet ugyanabban a megtalált klaszterben kell találjunk.

6. Diszkusszió

A 3. és 4. táblázatok összevetéséből kiderül, hogy minden tekintetben alkalmasabbnak bizonyult a hasonlósági gráf építése során csupán a dokumentumonkénti

4. táblázat. Automatikus kulcsszavak átfedése alapján épített hasonlósági gráf particionálásának eredményei.

	Közösségek száma	Modularitás	Pontosság	V_1
Max	12	0,689	0,303	0,365
Min	15	0,766	0,344	0,406
Átlag	14	0,763	0,300	0,391
Szorzat	16	0,805	0,303	0,353
Harmonikus közép	18	0,777	0,350	0,407
Dice	15	0,712	0,288	0,365
Jaccard	17	0,720	0,329	0,373

legjobb tíz kulcsszóra támaszkodni, szemben azzal a megközelítéssel, amikor a dokumentum összes n-gramjai közül a tíz legmagasabb tf-idf értékűvel lettek jellemezve az egyes dokumentumok. A kulcsszó alapú megközelítés javára írható az is, hogy annak használata mellett a kialakuló közösségek száma közelebbi volt az MSzNy korpuszban beazonosított 27 önálló téma mennyiségéhez.

Mérési eredményeink alapján a dokumentumpárok hasonlóságának súlyozására az átfedésben álló kulcsszavak jóságmértékének harmonikus közepet használó eljárás mondható a legjobbnak (mind az egyszerű n-gramokon, mind pedig a kulcsszavakon alapuló módszer esetében). Ez egyébként megegyezik előzetes várakozásainkkal, hiszen más megközelítések vagy egyáltalán nem hasznosítják a kulcsszavak jóságának mértékét (pl. Dice-együttható), vagy valamilyen értelemben túl szigorúnak (pl. Min), esetleg túl megengedőnek (pl. Max) mondhatók.

További előnyként mutatkozik, hogy a szótár mérete – vagyis azon kifejezések száma, amelyek a dokumentumok közötti összeköttetésekért felelhetnek azzal, hogy legalább egy dokumentumban szerepelnek – a kulcsszavakat figyelembe vevő módszer esetében 2079, míg a dokumentumokban szereplő n-gramokat alapul vevő eljárás esetében ennek több, mint 65-szöröse, 133754 volt.

Ez utóbbi érték természetesen nem azon kifejezések száma, amelyek mind felelősek lehettek két dokumentum közötti hasonlóság megállapítására az n-gram alapú modellben, hiszen dokumentumonként legfeljebb tíz kifejezés lehetett csupán ilyen, a korpusz általunk vizsgált részét alkotó dokumentumok száma pedig kevesebb, mint 400 volt. Ugyanakkor ahhoz, hogy a dokumentumonkénti legjobb tíz tf-idf értékű kifejezés meghatározható legyen, ismernünk kellett az összes, a korpuszban leírt kifejezéssel kapcsolatos előfordulási statisztikát. Ezzel szemben a kulcsszavak meghatározása aktuálisan mindig csak egy dokumentum alapján történt esetünkben, vagyis a szótárt képző kifejezések meghatározása dokumentumonként, egymástól függetlenül történhetett, így minden dokumentum esetében elegendő volt csupán az azt leginkább jellemző tíz kulcsszót tárolni.

7. Konklúzió és további munka

Jelen munkában az MSzNy cikkarchívumának automatikus kulcsszavazását és a kulcsszavazáson alapuló klaszterezését vizsgáltuk. A dokumentumokból épített

hasonlósági gráf particionálására, és így a témájukban koherens diszjunkt részkorpuszok detektálására alkalmasabbnak bizonyult az a megközelítés, amely az automatikusan meghatározott kulcsszavakkal jellemzi az egyes dokumentumokat, mint az n-gram alapú modell. A közös kulcsszóval rendelkező dokumentumok hasonlóságának számszerűsítésére pedig az átfedő kulcskifejezések kulcsszavazó modell által predikált valószínűségeinek felhasználása mutatkozott célravezetőnek (szemben pl. az egyszerű tf-idf mutató használatával).

Munkánk során elkészült a korpusz klaszterezésének egy interaktív online vizualizációja is, amely elérhető a `rgai.inf.u-szeged.hu/DocViewer` oldalon.

A dokumentumok kulcsszavai, illetve a klaszterek hasznos segítséget nyújthatnak számos további (pl. információ-visszakereső) alkalmazás számára, valamint az egyes részkorpuszok (közösségek) méretének változásának időbeli dinamikájának vizsgálatán keresztül lehetőség nyílik a különböző részterületek fontosságának alakulásának monitorozására, trendkövetésre, melyeket a jövőbeli kutatásaink során mélyebben tervezünk vizsgálni.

Köszönetnyilvánítás

Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

1. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: ACM DL. (1999) 254–255
2. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers. ICADL'07, Berlin, Heidelberg, Springer-Verlag (2007) 317–326
3. Turney, P.: Coherent keyphrase extraction via web mining. In: Proceedings of IJCAI '03. (2003) 434–439
4. Berend, G.: Opinion expression mining by exploiting keyphrase extraction. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, Asian Federation of Natural Language Processing (2011) 1162–1170
5. Farkas, R., Berend, G., Hegedűs, I., Kárpáti, A., Krich, B.: Automatic free-text-tagging of online news archives. In: Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence, Amsterdam, The Netherlands, IOS Press (2010) 529–534
6. Ding, Z., Zhang, Q., Huang, X.: Keyphrase extraction from online news using binary integer programming. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, Asian Federation of Natural Language Processing (2011) 165–173
7. Berend, G., Farkas, R.: Kulcsszókinyerés magyar nyelvű tudományos publikációkból. In: MSzNy 2010 – VIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem (2010) 47–55

8. Gupta, S., Manning, C.: Analyzing the dynamics of research by extracting key aspects of scientific papers. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, Asian Federation of Natural Language Processing (2011) 1–9
9. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In Tanács, A., Vincze, V., eds.: MSzNy 2013 – IX. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem (2013) 368–374
10. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69**(2) (2004) 026113+
11. Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: Maximizing Modularity is hard. (2006)
12. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10) (2008) P10008+
13. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). (2007) 410–420

Információorientált dokumentumosztályozás a magyar Wikipédián

Subecz Zoltán[1], Farkas Richárd[2]

[1] Szolnoki Főiskola
5000 Szolnok, Tiszaligeti sétány 14.
subecz@szolf.hu
[2] Szegedi Tudományegyetem, Informatikai tanszékcsoport
Szeged, Árpád tér 2.
rfarkas@inf.u-szeged.hu

Kivonat: Az *információorientált dokumentumosztályozás* egy olyan speciális többcímkés dokumentumosztályozási feladat, ahol az osztályozás nem a dokumentum egészének témája, hanem a dokumentumban található speciális információ alapján történik. Az ilyen típusú feladatokat általában úgy oldják meg, hogy ún. indikátorkifejezéseket keresnek a szövegben, majd analizálják azok szövegkörnyezetét, hogy kiszűrjék a hamis pozitív találatokat [1]. Ebben a munkában használunk egy módszert a *lokális tartalommódosítók* gépi tanulására. A módszer csak dokumentumszintű tanító címkéket használ fel. A tartalommódosítókat általánosan kezeljük, egy adott nyelvi jelenség detektálása helyett (pl. tagadás). Egy rendszerbe integráljuk a dokumentumosztályozást és a tartalommódosulás felismerését. Munkánkban magyar nyelvű Wikipédia-szócikkeket dolgoztunk fel ezzel a módszerrel. Az angol nyelvű szövegekhez használt nyelvi elemzőket helyettesítettük magyar nyelvre kidolgozott elemzőkkel. A cikkben vizsgált fő kutatási kérdés az, hogy a magyar nyelvi elemzők mennyiben járulnak hozzá a feladat megoldásához.

1 Bevezető

Munkánkban olyan többcímkés szövegosztályozási feladatot oldottunk meg, ahol a célosztályok a szövegből kinyerhető speciális információval állnak kapcsolatban és nem a dokumentum általános témájával. Ebben a feladatban a célinformáció a dokumentum egy egyedének, például egy személynek valamilyen tulajdonsága, de a feladat a dokumentum szintjén való osztályba sorolás. Ehhez hasonló feladat például a következő:

Páciensek dohányzási szokása egy gyakori megjegyzés a klinikai kórlapok szöveges részében [2]. Ebben az esetben a feladat speciális információ megtalálása a szövegben, pl. az adott páciens dohányzik, dohányzott a múltban, vagy egyáltalán nem dohányzott, de végül az alkalmazásnak a dokumentumot (páciens) kell osztályba sorolnia.

Munkánkban labdarúgókkal kapcsolatos Wikipédia-szócikkeket dolgoztunk fel. Ezekben a cikkekben a labdarúgókról – sok más egyéb mellett – leírják, hogy melyik

csapatban játszottak eddig. Alkalmazásunknak a cikkeket ezek alapján kell osztályokba sorolni.

Ezeknél a feladatoknál a célinformáció csak megemlítésre kerül a dokumentumban, a dokumentum nagy része az információkinyerési feladat szempontjából nem lényeges. Ezzel ellentétben az általános szövegosztályozási feladatoknál a cél a dokumentumok teljes tartalma alapján történő osztályozás. Ez nem egy megszokott információkinyerési feladat, mivel a cél a dokumentumok osztályozása, és a tanító adathalmaz is csak ezen a szinten áll rendelkezésre. Tehát ez a speciális feladat az információkinyerés és a dokumentumosztályozás között helyezkedik el. Speciális megközelítést igényel a megoldása és *információorientált dokumentumosztályozásnak* nevezzük a későbbiekben.

Korábbi munkák [2,3,4] bemutatták, hogy az információorientált dokumentumosztályozás hatékonyan megvalósítható ún. *indikátorkifejezések* kézi vagy statisztikai összegyűjtésével. Azonban ezek a munkák megmutatták az indikátorkifejezések lokális *szövegkörnyezetének* vizsgálatának fontosságát. Például a dohányzási szokás felismeréséhez néhány indikátor szó (pl. *dohányzik*, *cigaretta*) elégséges, viszont ezek környezetét meg kell vizsgálni ahhoz, hogy megállapítsuk, hogy a szerepük megváltozott-e (pl. hogy tagadva, vagy múlt időben vannak-e). Például:

A páciens azt mondta, hogy nem dohányzik.
A páciens 5 éve még dohányzott

A cikkben vizsgált fő kutatási kérdés az, hogy a magyar nyelvi elemzők mennyiben járulnak hozzá a feladat megoldásához. Először az angol szövegekre készült alkalmazást futtattuk magyar szövegeken is. Majd megnéztük, hogy a magyar nyelvű előfeldolgozási rész mennyiben javította a módszer helyességét. Általánosságban azt kerestük, hogy milyen nyelvi sajátosságokra kell megoldást találnunk a magyar nyelvű szövegek feldolgozásához. Például a magyar nyelv szabad szórendje miatt azt feltételeztük, hogy a függőségi elemzés kiaknázása fontosabb a magyar nyelvű szövegeknél, mint angolban.

Munkánkban a következő tapasztalatokra jutottunk: Az angol nyelvre kidolgozott programot változtatás nélkül alkalmazva a magyar szövegen a várakozásoknak megfelelően gyenge eredményeket kaptunk. Azonban magyar nyelvi elemzők (lemmatizáló, szófaji egyértelműsítő, függőségi elemző) alkalmazásával az eredmények jelentősen javultak, és megközelítették az eredetileg angol szövegekre kapott értékeket.

2 Tartalommódosító detektálás

Munkánkban egy egyszerű, de hatékony módszert alkalmazunk az információorientált dokumentumosztályozási feladat megoldásához [1], ami lehetővé teszi az indikátorkifejezések jelentésének megváltozásának észlelését. Ezeket *tartalommódosítóknak* nevezzük, és ezek azonosításának a feladatát pedig *tartalommódosító detektálásnak* (Content Shift Detection: CSD).

A rendszer bemenete egy dokumentumszinten címkézett tanító korpusz. Módszerünk kiválogatja az indikátorkifejezéseket és tanítja a CSD-t párhuzamosan. A helyi tartalommódosítókra fókuszálunk, és csak az indikátorkifejezést tartalmazó mondatokra koncentrálunk. Alapfeltevésünk az, hogy a CSD-t tudjuk az által tanítani, hogy megkeressük a tanító adathalmazban az indikátorkifejezések hamis pozitív (FP) előfordulásait.

A feldolgozott korpusz a Wikipédián szereplő labdarúgó-játékosok voltak. Ez egy információorientált dokumentumcímkézési feladat, mert a sportolóhoz tartozó cikkben csak röviden van megemlítve, hogy melyik csapatoknál játszott. A feladat többcímkés dokumentumosztályozás, mert egy labdarúgóhoz általában több csapat is tartozik.

Példák a magyar korpuszból az indikátorkifejezések tartalommódosulására: Mindkét példában a labdarúgó csak a Videoton FC csapatban játszott.

1. példa:
Bátyja Szakály Péter, a Debreceni VSC játékosa.

Ha csak a *Debreceni VSC* indikátorkifejezést nézzük, akkor azt gondolhatnánk, hogy a labdarúgó a Debreceni VSC játékosa. De a szövegkörnyezetből látszik, hogy nem ő, hanem a bátyja játszik abban a csapatban.

2. példa:
Bemutatkozó mérkőzése hazai pályán az MTK ellen volt.

Az MTK indikátorkifejezés arra utalna, hogy a játékos az MTK-ban játszik. De a szövegkörnyezetet megvizsgálva látszik, hogy az MTK csapat ellen játszottak.

Példa indikátorkifejezés kiválasztásra: A Győri ETO FC csapatához a következő indikátorkifejezéseket választotta ki az alkalmazás: [győr, a rába, a győri].

A Wikipédia-kategóriákon taníthatjuk az osztályozót ismeretlen szövegek címkézésére. Az így kapott modellel például adhatunk olyan csapatneveket is egy-egy szócikkhez, amelyik nincs feltüntetve, azaz automatikusan javíthatjuk a Wikipédia kategória-hozzárendelését.

Általában az információorientált dokumentumosztályozásnál a tanító példák rendelkezésre állnak (pl. Wikipédia-kategóriák), így nem kell kézzel annotálni a szövegeket. Ez egy nagy előnye ennek a módszernek.

Ha a dokumentumcímkék rendelkezésre állnak tanítási időben, akkor egy iteratív módszert használhatunk a CSD és az indikátorszelekció együttes tanítására. A tanítási fázisnak két kimenete van: az indikátorkifejezések halmaza és a CSD. A CSD egy bináris függvény, amely meghatározza, hogy egy indikátorkifejezés jelentése egy adott szövegkörnyezetben módosult-e. Azok a jó indikátorkifejezések, amelyek utalnak a hozzájuk tartozó osztálycímkére. Az iteráció minden lépésénél minden címkéhez kiválasztjuk az indikátorkifejezéseket a dokumentumhalmaz aktuális állapota alapján. Az indikátorkifejezés környezete segítségével tanítjuk a CSD-t, a hamis pozitív (FP) indikátor találatok a pozitív (módosított jelentés), míg a valódi pozitív (TP) találatai (nem módosított jelentés) a negatív példák a CSD számára. A tanult CSD-t alkalmazzuk a kiinduló adathalmazra, és töröljük a kiinduló dokumentumokból azokat a szövegrészeket, amelyeket a CSD módosultnak jelölt. Minél jobbak az indikátoraink,

annál jobban lehet tanítani a CSD-t. Egy ilyen tisztított dokumentumhalmazt használva jobb indikátorokat tudunk kiválasztani. Az iterációs lépéseket egy adott konvergenciakritériumig végezhetjük. A munkánkban három iterációt alkalmaztunk, mert a korábbi kísérletek azt mutatták, hogy későbbi iterációk már nem javítanak szignifikánsan az eredményeken [1].

3 A korpusz bemutatása

A feldolgozott korpusz a Wikipédián szereplő labdarúgó-játékosok voltak. Minden Wikipédia-szócikk végén megtalálható, hogy az adott cikk milyen kategóriákhoz tartozik.

Például Nyilasi Tiborhoz a következő kategóriák vannak rendelve: *Magyar labdarúgók, Labdarúgó-középpályások, A Ferencváros labdarúgói, Az FK Austria Wien labdarúgói, Magyar bajnoki gólkirályok, Magyar labdarúgóedzők, Az FTC vezetőedzői, Az év magyar labdarúgói, Az 1978-as világbajnokság labdarúgói, Az 1982-es világbajnokság labdarúgói, Várpalotaiak, 1955-ben született személyek.*

Egy ilyen kategória a **Magyar labdarúgók** is.

A Wikipédia aktuális állapotát rendszeresen lementik XML formátumú ún. DUMP fájlba (2 GB).[1] Ezen fájl letöltése után kiválogattuk azokat a szócikkeket, amelyekhez a Magyar labdarúgók kategória is volt rendelve (2069 szócikk).

Először elvégeztük a Wikipédia-szövegek tisztítását, eltávolítottuk a feladathoz nem tartozó részeket a szövegekből. Ennek és még további tisztítási lépéseknek a segítségével az XML-dokumentumból elkészítettünk egy sima text formátumú szöveget, amely már csak a szócikkek szöveges részét tartalmazza.

Készítettünk egy szövegfájlt, amelybe kigyűjtöttük minden játékoshoz, hogy mely csapatokban játszott. Ezek alapján kigyűjtöttük, hogy melyik csapathoz hány játékos tartozik, és kiválasztottuk a legismertebb klubokat: az első tíz csapatot, amelyikhez a legtöbben tartoznak. (1. táblázat, 2. oszlop)

A játékosoknál a csapatnevekre nem egységes volt a hivatkozás a kategóriáknál sem. Például a Vasas SC névvel hivatkoznak a Budapesti Vasas SC csapatra, vagy DVTK névvel a Diósgyőri VTK csapatra. Minden csapatra megtalálható a Wikipédián, hogy milyen neveken szerepelt a múltban. Ezek alapján egységesítettük ennek a tíz csapatnak a múltbeli hivatkozásait. Így kaptuk a 1. táblázat 3. oszlopában látható előfordulásokat.

A korpuszt véletlenszerűen tanító és kiértékelő részekre bontottuk: 300 dokumentumot tettünk a kiértékelő részbe, a maradékot pedig a tanító részbe.

Korpuszhibákat az osztályozás első eredményeinek elemzése közben is észrevettünk. A nem releváns (FP) és a nem szereplő releváns (FN) osztályozási eredményeket összevetve a korpusz adataival azt tapasztaltuk, hogy a magyar Wikipédia szövegeinél sok helyen nincs jól megadva, hogy egy adott játékos melyik csapatoknál játszott. Volt olyan szócikk, ahol a szövegben szerepelt, hogy melyik csapatban játszott, de nem szerepelt a címkénél. És volt olyan, hogy a címkénél szerepelt, de a szövegben nem.

[1] http://meta.wikimedia.org/wiki/Data_dumps

Ez jelentősen befolyásolta a kiértékelés megbízhatóságát, ezért a korpuszt manuálisan végignéztük és javítottuk a címkéket a nem megfelelő helyeken.

1. táblázat: Az első 10 csapat kiválasztása.

Csapat neve	Az eredeti előfordulások	Az egységesített előfordulások	További korpuszjavítás
Ferencvárosi TC	425	433	363
MTK Budapest FC	314	335	260
Újpest FC	319	330	249
Budapest Honvéd FC	250	269	185
Budapesti Vasas SC	201	252	182
Győri ETO FC	201	203	139
Videoton FC	150	153	108
Debreceni VSC	136	142	110
Diósgyőri VTK	129	135	97
Szombathelyi Haladás	105	108	84
Összesen	**2230**	**2360**	**1777**

581 címkét kellett javítani a teljes korpuszon. Ez a javítás a korpusz csökkenését is eredményezte. Így 1015 tanító dokumentum és 255 kiértékelő dokumentum maradt, azaz átlagosan 1,4 címke tartozik egy dokumentumhoz (1. táblázat, 4. oszlop).

4 Magyar elemző modulok

Az angol szövegeket feldolgozó alkalmazáshoz saját modult illesztettünk, amely elvégzi a mondatokra és szavakra bontást és választhatóan a lemmatizálást is. Ehhez a magyarlanc programcsomagot használtuk fel [6]. A magyarlanc programcsomag[2] magyar nyelvű szövegek alap, nyelvi elemzésére szolgál. A csomag tisztán JAVA nyelvű modulokat tartalmaz, ami biztosítja a platformfüggetlenséget és a nagyobb rendszerekbe (például webszerverek) történő integrálhatóságot. A csomag magában foglal egy magyar nyelvre adaptált mondat- és tokenszegmentálót, illetve egy szófaji elemzőt és egy függőségi elemzőt [6]. A szófaji elemző (lemmatizáló és POS-tagger) a Stanford POS-tagger[3] egy módosított változata, amely az ismeretlen szavakra a morfológiai elemző által adott lehetséges elemzéseket használja fel. Azon szóalakok esetén, amelyek nem szerepelnek a tanító adatbázisban, egy morfológiai elemző meghatározza a lehetséges elemzések halmazát, majd a szófaji egyértelműsítő modulnak ezen halmazból kell választania [5].

Az angol nyelvre készített program felhasználta a MorphAdorner csomag[4] mondatokra, szavakra bontó és lemmatizáló modulját, valamint a Stanford csomag

[2] http://www.inf.u-szeged.hu/rgai/magyarlanc
[3] http://nlp.stanford.edu/software/tagger.shtml
[4] http://morphadorner.northwestern.edu/

tokenizáló és PCFG parser[5] modulját. A magyar nyelvre készített alkalmazás ezeket a magyarlanc [6] programcsomaggal helyettesíti. Ez végzi el a mondatokra és szavakra bontást és a lemmatizálást is. A program függőségi elemző része az adott mondathoz meghatározza annak nyelvtani struktúráját, és ez alapján minden szóhoz meghatározza, hogy melyik szóhoz kapcsolódik alárendelve nyelvtanilag, és hogy milyen szerepet tölt be a kapcsolatban. Az elemzési fa csomópontjaiban a szavak állnak, az ágai pedig a közöttük lévő kapcsolatok. Az elemzőfa kiinduló pontja (Root) mindig egy ige. Az alkalmazásba a Bohnet-parser függőségi elemzőt integrálták be.

Egy további javítást végeztünk el a magyar szövegek mondatokra bontó részén: Azokat a mondatokat, amelyek számmal kezdődtek, nem választotta szét az előző mondattól a tokenizáló. Mivel a korpuszon sok mondat kezdődik évszámmal, így ezeknél még külön két mondatra bontottuk azokat. Eddig a mondatok száma 10614 volt, ezzel a javítással 14741 lett. Látszik, hogy ez sok mondatot érintett. Ezen kívül voltak olyan mondatok, ahol az első betű közvetlenül az előző mondatot lezáró pont után következett szóköz nélkül. Ezeken a helyeken be kellett szúrni egy szóközt, hogy két külön mondatra válassza azt a tokenizáló.

5 Az indikátorkifejezések (jellemzők) kigyűjtése

Az indikátorkifejezések szavak sorozata, amelyek jelenléte utal a pozitív osztályra. 1, 2 vagy 3 hosszúságú kifejezéseket választottunk ki. Az indikátorkifejezések kiválasztására több fajta algoritmus áll rendelkezésre. A munkánkban egy jellemzőkiértékelésen alapuló mohó algoritmust alkalmaztunk az indikátorkifejezések kiválasztására az összes kifejezés halmazából. Az indikátorkiválasztási célunk az volt, hogy az összes pozitív dokumentumot kiválasszuk, miközben minél kevesebb nem releváns esetet kapjunk. A mohó algoritmus iterációnként kiválasztja a legjobb kifejezést egy jellemzőkiválasztó metrika alapján [1].

6 Dokumentumosztályozás

A Wikipédián minden játékoshoz ki van gyűjtve, hogy milyen csapatokban játszott eddig. Ezt a kigyűjtést címkézésnek is tekinthetjük. Puskás Ferencnél a következő csapatok vannak feltüntetve: *Budapest Honvéd FC, Real Madrid CF*.

Mivel egy játékoshoz általában több klub is tartozik, ezért ez egy többcímkés osztályozási feladat. Vizsgálatunkban nem foglalkoztunk címkék közötti függőségekkel, így a többcímkés osztályozást bináris (pozitív vagy negatív) osztályozásra vezettük vissza. Így a többcímkés osztályozó modellünk a bináris osztályozók egy halmaza. Az osztályozó nálunk az indikátor-előfordulást vizsgálja. Ha találtunk egy nem módosított indikátort a szövegben, akkor az osztálycímkét a dokumentumhoz rendeljük.

[5] http://nlp.stanford.edu/software/lex-parser.shtml

Abból a feltevésből indultunk ki, hogy míg az indikátorkifejezések osztályfüggők, addig a tartalommódosítók tanulhatók osztálytól függetlenül is. Ez a megközelítés elég sok tanító adatot ad a tartalommódosulás detektálásának tanításához.

A CSD egy bináris osztályozó, amely az indikátort tartalmazó mondat alapján eldönti, hogy módosított-e az indikátorok tartalma. A bináris tanuló szózsák és szintaktikai kapcsolat alapú jellemzőkön alapul [1].

7 Eredmények

7.1 Kiértékelési metrikák

A különböző módszerek kiértékeléséhez az alábbi metrikákat használtuk:

- **Dokumentumosztályozás CSD nélkül:** Ebben az esetben csak az indikátorszavak alapján végezzük el az osztályozást. Azaz ha talál egy indikátort a dokumentumban, akkor egyből a dokumentumhoz rendeli a címkét. A kiértékelési metrikák az egyes címkékhez tartozó bináris osztályozási feladat pozitív osztályának pontosság, fedés és F-értékei. A végső értékek az egyes címkék mikroátlagai.
- **Dokumentumosztályozás CSD-vel**: Ugyanaz, mint az előző, a tanított CSD alkalmazásával. A dokumentumban található minden indikátor esetén megkérdezzük a CSD-t, hogy módosult-e az indikátor. Ha van olyan indikátor a dokumentumban, amely nem módosult, akkor a dokumentumhoz rendeli az adott címkét.
- **CSD önállóan**: Itt címkétől független magát a CSD-t értékeljük ki. Azt mérjük, hogy hányszor jelezte egy indikátor-előfordulást módosítottnak a CSD, amikor tényleg az volt. Tehát egy releváns találat (TP) az az indikátor-előfordulás, amelyre a CSD igent mond, és a szóban forgó címke nem szerepel a dokumentum címkéi között (azaz ez a környezet módosított). Metrikának a „módosult” osztály pontosság, fedés és F-értékeit használjuk.

7.1 Vizsgálat az eredeti programmal az angol korpuszon

Az **angol nyelvű szövegeken** az eredeti program lemmatizálással, bigramokkal és függőségi elemzés nélkül a 2. táblázat 1. sorában látható eredményeket éri el. Látjuk, hogy a CSD szignifikánsan javította a dokumentumosztályozás eredményét. Megjegyezzük, hogy a CSD három információorientált dokumentumosztályozási feladaton is szignifikánsan jobbnak bizonyult, mint a standard dokumentumosztályozási módszerek [1], ezért a magyar nyelvű kísérleteknél csak ezzel foglalkoztunk. Ez volt a kiinduló rendszerünk. Innen vizsgáltuk a magyar szövegekre kidolgozott programrészek használatának hatását.

2. táblázat: A tartalommódosulás-detektálás eredményei pontosság/fedés/F-érték formátumban.

	Dokumentumosztályozás CSD nélkül	Dokumentumosztályozás CSD-vel	CSD önállóan
az angol korpuszon	0.851/ 0.854/0.853	0.911/0.855/0.882	0.965/0.454/0.617
a magyar korpuszon	0.696/ 0.857/0.768	0.724/0.849/0.782	0.909/0.136/0.236
magyar tokenizáló és lemmatizáló	0.728/0.949/0.824	0.760/0.944/0.842	0.913/0.152/0.260
indikátort követő szavakkal	**0.728/0.949/0.824**	**0.792/0.944/0.861**	**0.937/0.316/0.473**
bigramokkal, lemmatizálás nélkül	0.727/0.842/0.780	0.783/0.844/0.812	0.847/0.297/0.440
unigramokkal lemmatizálással	0.700/0.972/0.814	0.773/0.954/0.854	0.896/0.320/0.472

7.2 Az eredeti program tesztelése a magyar szövegeken

Ezek után az angol nyelvű szövegre kidolgozott programot teszteltük a **magyar javított szövegen** (2. táblázat 2. sora). Itt **az eredeti programon nem változtattunk**, csak az angol nyelvű korpusz helyett a magyar nyelvűt használtuk. Az eredmények várható módon jóval alatta maradtak az angol szövegeken kapott értékeknek (pontosság, fedés, F-érték értékek jelentős csökkenése). Hiszen itt még az angol nyelvre kidolgozott tokenizáló és lemmatizáló programokat használtuk.

7.3 A program tesztelése a magyar tokenizálóval és lemmatizálóval

A programba beépítettük a magyarlanc tokenizálóját és lemmatizálóját. A következő tesztelést ebben a környezetben végeztük el. Először lemmatizálással és bigramokkal vizsgáltuk az alkalmazás működését. A 2. táblázat 3. során látjuk, hogy ez 6 százalékpontnyi javulást eredményez.

7.4 Az indikátorkifejezéseket követő szavak vizsgálata

Az angol nyelvű program a CSD jellemzőinek az indikátorkifejezések előtti szavakat gyűjtötte ki (szózsák modell). A magyar nyelvűnél a szövegeken azt láttuk, hogy az indikátorkifejezések utáni szavak is módosíthatják a szerepüket.

3. példa:
Első gólját 2000. április 1-jén a Győri ETO FC ellen szerezte.

Itt a csapatnév: Győri ETO FC, indikátorkifejezés: Győri ETO. Ha csak az indikátorkifejezés előtti szavakat vizsgáljuk a mondatban, akkor abból arra is következtethetünk, hogy a Győri ETO FC csapatban játszott. De ha kigyűjtjük az indikátorkifejezés utáni szavakat is, akkor abból egyértelműen kiderül, hogy ezen a mérkőzésen nem a Győri ETO FC csapatban játszott.

Ezért a **mondat többi szavát is kigyűjtöttük** az osztályozáshoz. A 2. táblázat 4. során látjuk, hogy ennek hatására a CSD fedése 15 százalékponttal javult, ami a teljes dokumentumcímkézési feladat 2 százalékpontnyi javulását vonta magával.

7.5 Indikátorkifejezések

Megvizsgáltuk az osztályozó működését **unigramokkal** és **lemmatizálás nélkül** is. Az eredmények a várakozásoknak megfelelően alatta maradtak a bigramokkal és lemmatizálással való tesztelésnek (2. táblázat 5. és 6. sora).

A további kísérleteket a legeredményesebb (**Bigramokkal, lemmatizálással**) konfiguráció alkalmazásával hajtottuk végre.

7.6 Szintaktikai környezet

Eddig a CSD tanításakor a mondatban az indikátorkifejezés előtti és utáni szavakat használtuk fel (szózsák). Megvizsgáltuk az indikátorkifejezés szintaktikai környezetét is. A szózsák jellemzők mellé beillesztettük a mondatokra alkalmazott függőségi elemzésből kinyerhető jellemzőket is.

3. táblázat: A szintaktikai környezet vizsgálata pontosság/fedés/F érték formátumban.

	Dokumentum-osztályozás CSD-vel	CSD önállóan
Indikátortól a Root-ig	0.806/0.944/0.869	0.942/0.355/0.515
Lemma vizsgálata	0.811/0.941/0.871	0.928/0.376/0.536
Alany az útvonalon	0.810/0.938/0.870	0.945/0.376/0.538

Először minden indikátorkifejezéshez megkerestük és kigyűjtöttük a hozzá tartozó részfa szavait: az indikátorszótól a Root-ig tartó útvonalon az adott szót és a szülő csomóponttal való kapcsolatát. Ez azért fontos, mert a Root a mondat egy kiemelt szava, és az indikátorszótól a Root-ig lévő szavak erősen meghatározzák az indikátorszó szerepét. Ezeket is betettük az osztályozóba a **jellemző**k közé. Például ilyen jel-

legű kifejezéseket: DEPrln#MODE, vagy: DEPgovrln#ATT#ellen. A 3. táblázat 1. sorát a 2. táblázat 4. sorával összehasonlítva látjuk, hogy ezen jellemzők 4 százalékpontnyit javítottak a CSD-n, ami majdnem 1 százalékpontnyi javulást eredményezett a dokumentumosztályozási feladaton.

Az előző módszeren még annyit változtattunk, hogy a Root-ig tartó útvonalon nem a szót, hanem annak lemmáját tettük be (3. táblázat 2. sor). **Ez adta a vizsgálatunk legjobb eredményét**.

Az indikátorszótól a Root-ig végigmenve, ha valamelyik csomóponthoz alany (SUBJ) kapcsolódik, akkor az alanyt és a kapcsolat típusát felveszi az osztályozási jellemzők közé. Ez azért lehet fontos, mert a szócikkhez tartozó játékos gyakran a mondat alanya, így ezen tulajdonság kigyűjtése meghatározza az indikátorszóhoz való viszonyát (3. táblázat 3. sor). A pontosság/fedés/F-érték értékeken látjuk, hogy eredmények nem javultak az eddigi legjobb eredményhez képest.

9 Diszkusszió

A 2. és 3. táblázatok adatai alapján megállapíthatjuk, hogy a tokenizáló (2,9%), a lemmatizáló (további 3%) és az indikátort követő szavak kiválasztása (további 1,9%) jelentősen javította az eredményeket. A függőségi elemző használata is javította az eredményeket (további 1%).

Azt is megállapíthatjuk, hogy a magyar elemzőeszközök használatával hasonló eredményeket értünk el az angol eredményekhez képest. Azaz kijelenthetjük, hogy a feladatra hasonlóan jól működő megoldás adható magyar nyelven, mint angolra.

A táblázatok minden sorában látható, hogy a CSD használata jelentősen javította a dokumentumosztályozás eredményeit. A fedésértékek kis csökkenése mellett a pontosságértékek jelentős javulását látjuk (hamis pozitív találatokat szűrhetünk ki a CSD-vel).

10 Példák a CSD jellemzőterére

4. példa:

Az MTK labdarúgója volt, ahol három bajnoki címet és egy magyar kupát nyert a csapattal.

Indikátorszó: MTK

Szózsák tulajdonságok: labdarúgó, csapat, ahol, van, magyar, és, bajnoki, cím, egy, nyer, kupa, három

Szintaxisalapú tulajdonságok: DEPrln#ATT, DEPgov#volt, DEPrln#PRED, DEPgovrln#PRED#volt, DEPgov#labdarúgója, DEPgovrln#ATT#labdarúgója,

Itt a DEP tulajdonság a függőségi elemzésre utal, az *ATT, PRED* tulajdonságok az adott szó grammatikai függőségét jelölik az elemzési fában felette lévő szóhoz képest. Erre utal az *rln* jelzés is: a kapcsolat típusa. A *DEPgov* adja meg a felette levő szót.

5. példa:

*Bemutatkozó mérkőzése hazai pályán az MTK **ellen** volt, ahol tizenegy percet játszott.*

Indikátorszó: MTK

Szózsák tulajdonságok: pálya, ellen, az, hazai, játszik, ahol, mérkőzés, van, perc, tizenegy, bemutatkozó

Szintaxisalapú tulajdonságok: DEPrln#ATT, DEPgov#ellen, DEPgovrln#ATT#ellen, DEPrln#MODE, DEPgov#volt, DEPgovrln#MODE#volt

Mindkét példánál egy naiv rendszer pozitív címkézést adna. Az első példánál ez igaz is, de a második példánál ez nem releváns találat (FP) lenne. A CSD tanításával azt várjuk, hogy az osztályozó a második mondatnál negatív jelzést adjon. Ennél a példánál a CSD megtanulta, hogy a szózsák tartalmazza az *ellen* szót, ami módosítja az MTK indikátorszó tartalmát.

11 Összegzés

Ebben a cikkben az információorientált dokumentumcímkézési feladatokkal foglalkoztunk és gépi tanulási módszereket vizsgáltunk meg a helyi tartalommódosulás detektálásához. Empirikusan bizonyítottuk, hogy a nem releváns indikátorkifejezések felismerhetőek CSD tanításával. A tanított CSD nem használ semmilyen feladat-, vagy doménspecifikus ismeretet, csak dokumentumszintű annotált címkéket. Egy rendszerbe integráltuk a dokumentumosztályozást és a tartalommódosulás felismerést.

Munkánkban magyar nyelvű Wikipédia-szócikkeket dolgoztunk fel ezzel a módszerrel. Kiválasztottuk a magyar labdarúgók szócikkeivel kapcsolatos korpuszt, amelyet manuálisan javítottunk. Az angol nyelvű szövegekhez használt nyelvi elemzőket helyettesítettük magyar nyelvre kidolgozott tokenizáló, lemmatizáló és függőségi elemző modulokkal. A tokenizáló (2,9%), a lemmatizáló (3%) és az indikátort követő szavak kiválasztása (1,9%) jelentősen javította az eredményeket. A függőségi elemző használata is javította az eredményeket (további 1%). A magyar nyelvi modulok összesen így 8,9 százalékponttal (közel 40 százalékos hibacsökkentés) javították a dokumentumcímkézés hatékonyságát és az angol feladaton elért eredményekhez hasonló eredményt értünk el a magyar korpuszon.

Köszönetnyilvánítás

Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

1. Farkas, R.: Learning Local Content Shift Detectors from Document-level Information. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. John McIntyre Conference Centre, Edinburgh, UK (2011) 759–770
2. Uzuner, Ö., Goldstein, I., Luo, Y., Kohane, I.: Identifying Patient Smoking Status from Medical Discharge Records. Journal of American Medical Informatics Association, Vol. 15, No. 1 (2008) 14–24
3. Uzuner, Ö.: Recognizing obesity and comorbidities in sparse data. Journal of American Medical Informatics Association, Vol. 16, No. 4 (2009) 561–70
4. Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K. B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the ACL Workshop on BioNLP (2007) 97–104
5. Zsibrita J., Vincze V., Farkas R.: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 275–283
6. Zsibrita J., Vincze V., Farkas R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 368–374

Frame-szemantikára alapozott információ-visszakereső rendszer

Szőts Miklós[1], Gyarmathy Zsófia[1], Simonyi András[1]

[1]Alkalmazott Logikai Laboratórium
1022 Budapest, Hankóczy j. u. 7
{szots,simonyi}@all.hu, gyzsof@gmail.com

Kivonat: Egy olyan információ-visszakereső rendszert mutatunk be, amely kontrollált természetes nyelven megadható keresőkifejezésekhez keres hasonló jelentésű szövegrészt tartalmazó természetes nyelvű dokumentumokat. A rendszer frame-szemantikai elemzéssel előállítja a keresőkifejezés szemantikus reprezentációját, és azokat a dokumentumokat adja vissza találatként, amelyekben található olyan szövegrész, amelyhez a reprezentáció illeszthető. Cikkünkben ismertetjük a rendszer működését és az általa használt szemantikus reprezentációk, illetve erőforrások felépítését – elsősorban a frame-szemantika alkalmazására koncentrálva. Röviden kitérünk a még hátralévő feladatokra és a lehetséges további kutatási irányokra is.

1 Bevezetés

Az Alkalmazott Logikai Laboratórium és a Szegedi Tudományegyetem Informatikai Tanszékcsoportja, valamint Könyvtártudományi Tanszéke közös projektet (TECH_08_A2/2-2008-0092) indított az NFÜ támogatásával szemantikus információ-visszakereső rendszer kifejlesztésére. A tervezett projekt célja egy olyan, új elveken alapuló integrált keresőrendszer, a MASZEKER kifejlesztése, amely a keresést végző felhasználó szemantikai kompetenciáját az eddigieknél nagyobb mértékben kiaknázva teszi lehetővé a természetes nyelvi dokumentumtárakban (szövegekben) történő valóban *tartalmi* keresést. Egyszerűen szólva: a felhasználó jól formált frázisokkal, mondatokkal specifikálhatja, milyen tartalmú dokumentumokat keres. Terveinkről 2010-ben számoltunk be a VII. MSzNy konferencián [1], 2011-ben pedig az első prototípust mutattuk be [2]. A projekt 2012-ben lezárult. Eredményeinkről számol be ez az előadás.

A projekt során kifejlesztettük a szemantikai keresés technológiáját, és két rendszert: az egyiket angol nyelvű szabadalmi leírásokban, a másikat magyar nyelvű néprajzi anyagokban való keresésre. Itt a technológiát ismertetjük, a szabadalmi rendszert programbemutatón [3] ismerheti meg az érdeklődő. A magyar nyelven működő néprajzi keresőrendszert [4] ismerteti.

2 State of the art

Természetesen – mint annyi szakszó az informatikában – a „szemantikus információ-visszakereső rendszer" kifejezésben szereplő „szemantikus" is a lehető legkülönbözőbben értelmezhető. Sokan a szavak, szóösszetételek szintjén értelmezik: szavak közti jelentés-összefüggések feltárásával egészítik a ki a kulcsszó szerinti keresést. Ilyen a már elterjedt látens szemantika algoritmusa[1] (l. [5]) is. Elterjedőben van a keresők valamilyen ontológiához, tezauruszhoz való kapcsolása, ilyen alapon működik a magyar fejlesztésű, de nemzetközi hírnevet szerző HealthMash kereső is (l. http://www.weblib.com/products/healthmash). A MEDLINE-on működő KLEIO kereső (ismertetőt találhatunk [6]-ban) szintén ontológiákhoz van kapcsolva, de a névelem-felismerés (NER) technikáját is használja. A keresőkifejezésben megengedi, hogy a kulcsszavakhoz a felhasználó megadja annak besorolását, pl. *PROTEIN*:cat, amit a keresés pontosságának javítására használ. Mi azonban szemantikus keresés alatt olyan folyamatot értünk, amely összefüggő szövegrészek jelentése alapján ítél valamely dokumentumot relevánsnak.

A szemantikus keresők két nagy osztályba sorolhatóak (l. [7]): lehetnek statikusak vagy dinamikusak. A statikus keresők előre elkészítik a keresett honlapok, dokumentumok szemantikus reprezentációját, és felindexelik azokat; míg a dinamikusak a keresőkifejezés jelentésreprezentációját a keresés alatt elemzett szövegrészekre illesztik. Szintén gyakran használt osztályozási szempont az, hogy témafüggetlenek, vagy egy tématerületre specializáltak. Csak néhány keresőrendszert sorolunk itt fel, egy teljesebb áttekintés letölthető a www.maszeker.hu oldalról.

A HAKIA (l. [8]) általános célú, ontológiai szemantikára (l. [9]) alapozott, statikus keresőrendszer. Honlapok szövegei jelentésreprezentációjának alapján előre elkészíti a lehetséges kérdésekre adható válaszokat, amelyek közül az adekvátat a keresés közben csak ki kell választania. Inkább a tudáskinyerés területéhez tartozik, de a szemantikus keresés általában könnyen átfogalmazható tudáskinyerésre. A HAKIA egy erre a célra kifejlesztett, 8 500 fogalmat tartalmazó ontológiára támaszkodik. Ehhez csatlakozik egy kb. 100 000 szójelentést és több mint 1 000 000 szót tartalmazó szótár.

A Cognition (l. [10]) egy átfogó, szintén statikus természetesnyelv-feldolgozó keretrendszer, amely egy témafüggetlen kereső motort is tartalmaz. Több, egy-egy területre vagy dokumentumhalmazra specializált alkalmazása van, pl. a Wikipediára, illetve a MEDLINE abstracts-ra is kifejlesztettek egy-egy speciális keresőt. Ontológiája 7 500 fogalmat tartalmaz, amelyekhez 536 000 szójelentés kapcsolódik.

A Powerset a Cognitionhoz hasonló rendszer. Sok információnk nincs róla, mivel a Microsoft megvette, és beépítette a fejlesztés alatt lévő keresőjébe (l. [11]).

Az UpTake (l. [12]) egy utazási információkat szolgáltató kereső, amely több mint 5 000 honlapot indexelt fel. Jellegzetessége, hogy a felhasználóval folytatott párbeszédet támogat, azaz az általánosabb kéréstől a specifikusabb felé mozoghat a felhasználó. Azt tervezik, hogy rendszer alapjául szolgáló ontológiát tanuló algoritmusokkal bővítik.

A GoWeb (l. [13]) az élettudományokra specializált kereső. Természetes nyelvű kifejezést fogad el bemenetként, s egy tradicionális, kulcsszó szerinti keresés eredmé-

1 Részletes ismertetése letölthető a www.maszeker.hu honlapról.

nyeit veti alá szemantikus elemzésnek. Háttere a Gene és a MeSH ontológia. Az eredményhez ezeknek az ontológiáknak a releváns részleteit is megmutatja. E leírásból is kitűnik, hogy a GoWeb dinamikus kereső.

A MEDIE (l. [6], [15]) a már említett KLEIO-hoz hasonlóan a MEDLINE-on keres; azonban a KLEIO-hoz képest jelentős előlépés, hogy már szintaktikus és szemantikus elemzést alkalmaz az események kinyerésére. Egyelőre csak *alany-ige-tárgy* alakú kereső kifejezéseket kezel. [6] beszámol további kutatási irányokról, amelyek hasonlóak a mieinkhez.

3. A technológia áttekintő ismertetése

A kifejlesztett technológia szerinti szemantikus információ-visszakeresés folyamata a következő lépésekből áll:

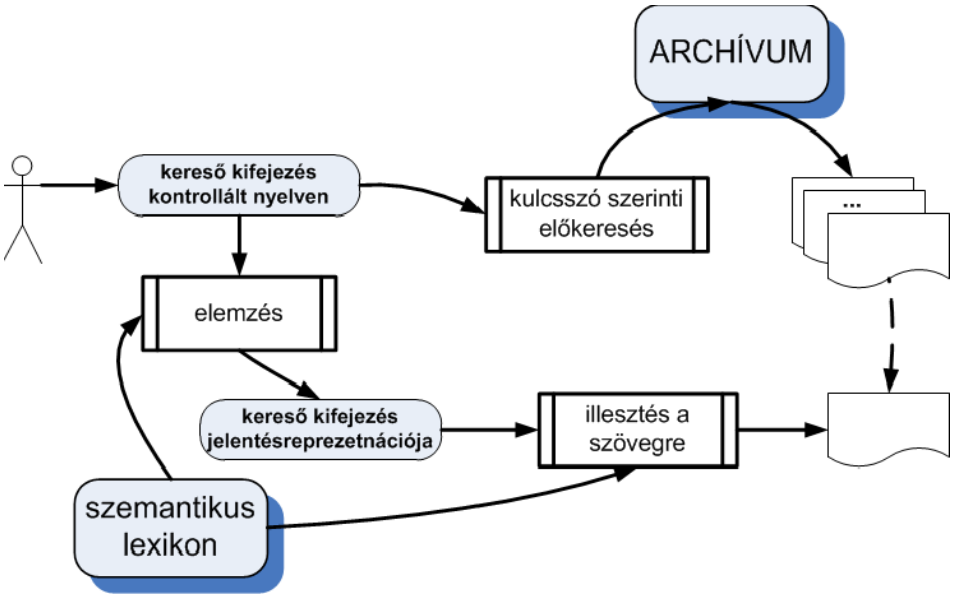
1. A felhasználó egy kontrollált nyelven megfogalmazott szöveget ad meg keresőkifejezésként.
2. Egy szintaktikai és kognitív szemantikai elemzési folyamat előállítja a keresőkifejezés jelentésreprezentációját. Az ehhez szükséges nyelvi és világtudást egy szemantikus lexikon írja le.
3. A keresőkifejezésben szereplő kifejezések és szinonimáik alapján egy kulcsszó alapú előkeresés kiválasztja a dokumentumokból azokat a szövegszegmenseket, amelyek a kulcsszavak előfordulása alapján találatok lehetnek.
4. Egy illesztési folyamat megy végbe, amely a keresőkifejezés jelentésreprezentációját ráilleszti azokra a szövegszegmensekre, amelyekben a szavak szerinti előkeresés találatai vannak, és az illesztés alapján elbírálásra kerül a keresőkifejezés és az adott szövegrészlet hasonlósága.
5. A felhasználó megkapja azon dokumentumrészletek listáját, amelyek a keresőkifejezés jelentéséhez leginkább hasonló jelentéssel bírhatnak. A lista a hasonlóság foka szerint rendezett.

A folyamat architektúráját az 1. ábra illusztrálja.

A fentiekből látható, hogy a technológia alapelve a következő:

A kontrollált nyelven megadott keresőkifejezésnek pontos jelentésreprezentációja generálható, a keresésnél a nyelvi és szemantikai elemzés bizonytalanságai heurisztikus szabályokkal oldatnak fel.

A következőkben a keresőkifejezés jelentésreprezentációjának előállítására koncentrálunk, e témában is a nyelvészeti kérdéseket hangsúlyozva.



1. ábra: A folyamat áttekintő architektúrája.

4 A jelentésreprezentáció előállítása

4.1 Szemantika, jelentés, jelentésreprezentáció

Míg a nyelvészet különböző területeinek vizsgálata a nyelven belül marad, a szemantika – legalábbis ahogy mi értjük – kilép belőle: a nyelvi konstrukciók jelentései általában nyelven kívüli entitások. A számítógépes nyelvészet területén az alkalmazás célja határozza meg, mit értünk jelentés alatt. Sőt, a jelentés fogalma háttérbe húzódik, a célnak megfelelő jelentésreprezentáció lesz fontos. Az információ-visszakeresés esetén olyan jelentésreprezentációt keresünk, amely a keresőkifejezésre és az ahhoz illő nyelvi konstrukciókra ugyanaz lesz. Kissé absztraktabban: a jelentésreprezentáció reprezentálja azt a szituációt, amelyet a nyelvi konstrukció jelenthet. A szituációk központi szerepéből adódóan a jelentésreprezentáció legfontosabb feladata a predikatív szavak és argumentumaik által alkotott szintaktikai egységek jelentésének pontos ábrázolása.

A fentiek alapján érthető, hogy a jelentésreprezentációk kialakítását davidsoni alapokon [15] kezdtük el, azaz az igék és az eseményszerűségeket jelentő főnevek jelentését reifikáljuk: maga az esemény egy token lesz, amelyhez a szereplőket szerepreláció kötik hozzá. A problémát a tematikus szerepek kiválasztása és ehhez kapcsolódóan a szemantikus lexikon szerkezetének meghatározása jelentette.

4.2 Szemantikus lexikon

A projektvezető legszívesebben a keresési feladathoz illő saját rendszert alkotott volna, tematikus megoszlásban más-más szerepkészlettel – de az erőforrás megalkotásával, feltöltésével járó munkát a projekten belül nem vállalhattuk, ezért meglévő erőforrás után néztünk. A követelmények a következők voltak:

- tartalmazzon vonzatkereteket, mégpedig szemantikus információval (megfelelő szerepekkel), nemcsak az igéknél, hanem a predikatív szavaknál általában, lehetőleg rugalmasan, ne adott számú kötött szerep legyen kiosztva;
- a szinonimahalmazok ne legyenek olyan szűkek, mint a WordNet esetében; elsősorban a következőkre gondoltunk:
  - az igékkel együtt szerepeljenek a hasonló jelentésű, eseményszerűséget jelentő főnevek (pl. a *treat* és *treatment*),
  - azok a szavak, amelyek csak a nézőpontban különböznek (pl. *give* és *get*), szintén együtt szerepeljenek;
- szerepeljenek benne szelekciós megszorítások (minél több, annál jobb);
- a szinonimahalmazokat relációk kössék össze, elsősorban az öröklődési reláció.

Az erőforrások áttekintése (l. [16]) a következő hármat találta:

- PropBank (http://verbs.colorado.edu/propbank/framesets-english/),
- VerbNet (http://verbs.colorado.edu/~mpalmer/projects/verbnet.html),
- FrameNet (http://framenet.icsi.berkeley.edu/).

A PropBank Arg1, Arg2 … jellegű szerepeket használ, amelyek csak nagyjából felelnek meg jelentéshordozó szerepeknek (Actor, Instrument stb.). A VerbNet – ahogy neve is mutatja – csak igéket tartalmaz, és kötött tematikusszerep-készletet használ. Így végül a FrameNetre esett a választásunk, amelynek egy verziója már letölthető volt.

A FrameNet, ahogy a neve is mutatja, a frame szemantika alapján épül ([17], [18]). A frame szemantika szerint a szavak jelentését egy szemantikai frame-en, avagy szemantikai kereten keresztül lehet megragadni, amely legtöbbször egy esemény, illetve szituáció leírása, a benne lévő szereplőkkel együtt. Például a főzés fogalmához mint kerethez hozzátartozik szereplőként, avagy „frame element”-ként az étel, az a személy, aki a főzést végzi, az ételt tartalmazó edény, valamint az energiaforrás is. Ez a jelentésmegközelítés tehát bizonyos szintű világtudást is magában foglal. Egy-egy frame-et több szó is felidézhet (ezeket a már frame-hez rendelt szavakat hívják “lexical unit”-oknak, azaz lexikai egységeknek), amelyek nem feltétlenül szinonimái egymásnak. Például a gyógyítás frame-et a *rehabilitáció* és az *ápoló* szavak is előhívhatják.

A FrameNet egy, a frame szemantikán alapuló lexikális adatbázis, amely a predikatív szavak leírását célozza meg. Az egyes predikatív szavakhoz megadja, hogy mely frame-eket hívják elő (az ekként jelentés-egyértelműsített szavak a “lexical unit”-ok), hogy milyen vonzatkeretekben (valence pattern, azaz valenciamintázat, amely nem csupán vonzatokat, hanem szabad határozókat is tartalmazhat) fordulhatnak elő, valamint hogy milyen relációban állnak velük az egyes vonzatkeretekben előforduló bővítményeik. Ez utóbbi relációt a “frame element” (frame-elem) fogalma fedi, amely egy adott frame-hez tartozó megfelelő szereplőt jelöl. A hagyományos tematikus szerepekkel ellentétben a frame-elemek frame-specifikusak (például a gyó-

gyítás frame-hez tartozik a Gyógyító frame element), azonban a frame-ekhez hasonlóan fennállhatnak közöttük öröklődési és egyéb kapcsolatok.

A FrameNet bővítése lexical unitokkal és valence patternekkel egy korpusz kézi frame szemantikai annotálásán keresztül, majd az annotált mondatok gépi feldolgozásával és adatbázissá alakításával történik. Ennek megfelelően, lévén korpuszalapú, időnként nyelvészetileg érthetetlen redundanciák vagy épp hiányok tapasztalhatók, és bizonyos mértékig (a gyakorisági hatást leszámítva) esetleges, hogy egy szó mely jelentése, illetve valenciamintázata kerül be az adatbázisba.

A FrameNet teljesíti legfontosabb követelményeinket, bár nem mindegyiket kielégítően:

- Az úgynevezett valence pattern-ök (VP-k, valencia mintázatok) jól leírják a régens és dependensei közti viszonyt. Nincs megkülönböztetve a vonzat és a szabad határozó. Nincsenek általános frame-elemek, első pillanatban úgy érezzük, hogy minden frame-hez frame-elemek egy külön rendszere tartozik. Azonban kimutatták [19], hogy az ugyanolyan névvel ellátott frame-elemek sok esetben azonosnak vagy közel azonosnak tekinthetőek.
- A frame szemantika szerint egy szó akkor tartozik egy frame-hez, ha a szóról az adott frame-re asszociálunk. Egy frame-hez tartozásnak nem feltétele a szófaj egyezése, így igék, főnevek, melléknevek, sőt, prepozíciók kerülhetnek egy frame-hez. Látható, hogy az egy frame-hez tartozó lexikális egységek kielégítik a keresés igényeit. Egyedül az az eltérés, hogy a nézőpontjukban különböző szavak külön frame-ekben vannak, viszont ezeket egy közös frame-hez köti a Perspective_on reláció. Mi összevonjuk ezeket a frame-eket egyetlen frame-be.
- Szemantikus megszorítások vannak a FrameNet-ben, de teljesen használhatatlanok. Az úgynevezett ontológiai szemantikus típusok meghatároznak egy kezdetleges fogalmi hierarchiát, ezek megadásával definiálják a szemantikus megszorításokat. Azonban az ontológiai szemantikus típusok semmiképp nincsenek összekapcsolva a frame-ekkel, ezért önmagukban használhatatlanok.
- A frame-eket relációk kötik össze, amelyek szerint öröklődnek a frame elemek is. Legfontosabbak az Inheritance, a Subframe és a Using relációk. Azonban a relációk értelmezése nem tiszta, használatuk nem következetes. Számunkra a leginkább fájó az volt, hogy az öröklődés nem követ valamilyen ontológiai elvet, hanem csak az egymásnak megfelelő frame elemekre figyel. Például: van négy „Cause_change of ….” kezdetű névvel ellátott frame, de egy sem öröklődik a „Cause_change” frame-ből[2].

Vannak a FrameNet-nek árnyoldalai is. Számunkra érthetetlen, hogy a VP-knél nem jelzik, hogy az ige aktív vagy passzív alakjához tartozik-e – ezt az annotált mondatokból kellett kiszűrnünk. Találtunk hibás besorolást, következetlen döntéseket.

Az irodalom (l. pl. [20]) arról tanúskodik, hogy a szóegyértelműsítés hibái különösen nagy kárt okozhatnak, mert a frame-specifikus szereprelációk miatt nagyobbat tévedhet a rendszer, ha rossz frame-hez köti az adott kifejezést, mint ha általánosabb szereprelációkat használnánk. Ezt azzal védjük ki, hogy a keresőkifejezés jelentésreprezentációjában az összes olyan frame-t felvesszük, amelyben a lexikális egységnek

2 Nem volt időnk alaposan végig vizsgálni ezeket, így nem tudjuk, hogy az öröklődés elve okozta ezt, vagy csak rosszul van alkalmazva.

van megfelelő VP-je, és a felhasználó egyértelműsíthet. A keresés során pedig a keresőkifejezés szabja meg a választott frame-et: ha egy szó tartozhat egy, a keresőkifejezés jelentésreprezentációjában szereplő frame-hez, odatartozónak vesszük.

A FrameNet másik hátránya az, hogy viszonylag kevéssé feltöltött. A 2011-es állapotban 960 frame és 11 600 lexikai egység volt, azonban ez utóbbiakból csak 6 800 teljesen annotált. Érthetően elsősorban a predikatív szavakat veszik fel, bár vannak kivételek. Ezért a nem predikatív szavakra nem megfelelő forrás, és mondanunk sem kell, hogy nem a tudományos jellegű szövegek szókincsét nyújtja.

Ezen okok miatt úgy határoztunk, hogy a szemantikus lexikon három rétegből álljon:

- a predikatív (vonzatkerettel rendelkező) szavak,
- a nem predikatív főnevek, melléknevek, határozók,
- tematikus csoportosításban az egyes szakterületek terminológiája.

A FrameNet csak az első réteg alapja lett, a másodiknak a WordNetet, a harmadiknak egy orvosbiológiai erőforrás, a MeSH[3] megfelelő szegmenseit vettük. A letöltött FrameNet-verzió csak alapja az első rétegnek, számos új lexikai egységet vettünk fel, és frame-ekkel, relációkkal, szemantikus megszorításokkal is gazdagítjuk. Bizonyos FrameNet-beli szelekciós megszorításokat helyettesíteni tudtunk WordNetre hivatkozókkal, felhasználva az ontológiai szemantikus típusoknak megfelelő WordNet szinonimahalmazokat.

4. 3 Jelentésreprezentáció

A jelentésreprezentációk ciklusmentes, címkézett irányított gráfok. A gráfban a predikatív szavaknak olyan csomópontok felelnek meg, amelyek az adott szónak megfelelő frame-mel vannak címkézve, míg a belőlük kiinduló élek a bővítményekre jellemző szintaktikus viszonynak megfelelő frame-elemmel címkézettek. Az élek a bővítmények reprezentációiba futnak. Tehát mi a frame elemeket frame-ek közt értelmezett relációknak tekintjük.

A jelentésreprezentáció előállításához természetesen morfoszintaktikai elemzés szükséges. Azonban csak olyan mértékben van szükségünk a szöveg szintaktikai szerkezetére, hogy a jelentésreprezentációt generálni lehessen. A jelentésreprezentáció sokszor igen kissé hasonlít a szintaktikus gráfhoz, ahogy a következő példán is láthatjuk. Tekintsük a következő két frázist: *a tablet containing aspirin* és *a tablet contains aspirin*. A jelentésreprezentációjuknak meg kell egyezni, mindkét esetben a *contain* ige a fej, míg az első frázisnál a *contain* a *tablet* bővítménye. Ebben az esetben a szintaktikus elemzésnek azt is be kell jelölnie, hogy a *tablet* a *contain* alanya. A szintaktikus elemzésről [21]-ben olvashatunk.

A jelentésreprezentáció előállításához a szavakhoz tartozó frame-eket és a szintaktikus viszonyokhoz tartozó frame-elemeket kell a szemantikus lexikonból kinyerni. Problémát az alternatívák nyilvántartása jelent; nemcsak az, hogy egy csomópont vagy él több címkét kaphat, hanem az is, hogy ezek összefüggenek. Egy szülő csomópont

3 Mivel a prototípus a gyógyszerek és kozmetikai szerek témaköréhez adaptált.

frame címkéjétől függ a belőle induló él frame-elem címkéje, és a gyermek csomópont címkéje függhet a hozzá vezető él címkéjétől (a szemantikus megszorítások miatt). Sőt, ugyanaz a csomópont lehet több szülő gyermeke is. A legcélszerűbb az lett volna, ha a szintaktikus elemzés és a jelentésreprezentáció generálása egyszerre történik, azonban történeti okokból nem így lett.

5. Keresés

A keresési folyamat az előkeresés által kiválasztott szövegszegmensek és a keresőkifejezés jelentésreprezentációja közti hasonlóság megállapításából áll. A feldolgozás szövegszegmensenként zajlik. Első lépésként meg kell vizsgálni, hogy a feldolgozandó szövegszegmens mely szavai tartozhatnak olyan frame-hez, illetve szinonimahalmazhoz, amely szerepel a keresőkifejezés szemantikai reprezentációjában. A második lépés annak meghatározása, hogy a keresőkifejezés elemeinek a szövegszegmensben megfelelő szavak lehetnek-e ugyanabban a szemantikai viszonyban, mint amelyben a keresőkifejezés szemantikai reprezentációjában nekik megfelelő elemek állnak. Ez a lépés három módon valósítható meg:

- Az adott szövegszegmens teljes szintaktikai elemzése alapján készül el a szegmens teljes jelentésreprezentációja, és ez a teljes reprezentáció kerül összehasonlításra a keresőkifejezés reprezentációjával.
- A szegmens teljes szintaktikai elemzése elkészül, és a keresés ezen a teljes szintaktikus gráfon működik, de csak azon szegmensbeli kifejezések szemantikai értéke kerül előállításra és vizsgáltatik meg, amelyek kapcsolatban állnak a keresőkifejezés reprezentációjában előforduló elemnek megfelelő frázissal.
- Teljes egészében a keresőkifejezés által vezérelt keresés megy végbe, tehát a szegmensnek a keresőkifejezés elemeinek megfelelő kifejezéseiből kiindulva részleges és párhuzamos szintaktikai és szemantikai elemzés történik, amely a grammatikai viszonyok megállapításával egy időben osztja ki a szemantikai szerepeket.

Az 1. megoldást valósítottuk meg. A megközelítés hátránya az, hogy nincsen megbízható módszer a nem kontrollált nyelven megfogalmazott szövegek pontos jelentésreprezentációjának előállítására. Viszont nincs is szükségünk a pontos jelentésreprezentációra, alkalmazhatjuk a „*jóindulatú olvasat*” elvét. Ez azt jelenti, hogy ha egy szó tartozhat olyan frame-be, synset-be, amely a keresőkifejezés jelentésreprezentációjában szerepel, vizsgálat nélkül elfogadja ezt az eljárás; hasonlóképpen, ha a bővítményekhez tartozó lehetséges frame-elemek közt van a keresőkifejezés jelentésreprezentációjában szereplő, az eljárás azt veszi figyelembe. Figyelni kell arra, hogy ugyanaz a csomópont a keresőkifejezés jelentésreprezentációjában előforduló több frame-hez/synset-hez is illik, - ekkor a hozzá tartozó frázis reprezentációját meg kell sokszorozni, mintha többször fordulna elő.

6. További teendők, kutatási irányok

A projekt sikeres volt: hatékony információ-visszakereső technológiát sikerült kidolgoznunk. A sikert egyrészt az biztosította, hogy csak a keresőkifejezés elemzésének kell pontosnak és egyértelműnek lennie, másrészt a frame-szemantikán alapuló jelentésreprezentáció. Természetesen a sikeresen megvalósított információ-visszakereső rendszer nem jelenti azt, hogy minden kutató-fejlesztő tevékenységet lezárhatunk. A találati halmaz pontossága megfelelőnek tűnik, viszont a fedést növelnünk kell. Vannak olyan fejlesztési feladatok, amelyek elméletileg tisztázottak, specifikálva is vannak (pl. a raising és control igék kezelése), ezekre itt nem térünk ki.

Van azonban két nyelvészetileg meg nem oldott probléma, amellyel szembetalálkoztunk:

- A felsorolások és koordinációk jellemzőek az igénypontok szövegére. A keresőkifejezés szerkesztésénél a felhasználónak meghatározott módon jeleznie kell a felsorolásokat, koordinációkat, a dokumentumok elemzésénél közelítő eljárást alkalmazunk.
- Az összetett szavak problémája sokkal fájóbb. Bonyolult kémiai kifejezéseket találtunk, sokat mi sem tudtunk értelmezni. A probléma a kifejezések zárójelezése. Az irodalomban sem találtunk nagy mennyiségű kézi annotálás nélkül implementálható megbízható megoldást problémánkra. [22] szerint a balra való zárójelezés a legjobb default, alapértelmezett eljárás – bár mi számos ellenpéldát láttunk.

A kognitív nyelvészethez tartozó frame szemantika alkalmazása nem volt céltudatos, hanem a FrameNet választása szinte észrevétlenül sodort minket bele. Természetesen a kognitív nyelvészet nagyon sok sajátossága a feladatunkhoz mellékes volt. Azonban most látjuk, hogy a frame szemantika beágyazási mechanizmusa alkalmas természetesnyelv-feldolgozási feladatok megoldáshoz. Az egyes szavakhoz tartozó frame-ek beágyazása elvezethet közös doménhez, így alkalmas egy-egy szövegszegmens témájának meghatározásához. Ez segítené a szóegyértelműsítést is.

Köszönetnyilvánítás

A kutatás az NFÜ által finanszírozott, MASZEKER kódnevű, TECH_08_A2/2-2008-0092 számú projekt keretében valósult meg.

Szeretnénk köszönetet mondani a projekt összes résztvevőjének is – nélkülük nem számolhatnánk be eredményeinkről.

Hivatkozások

1. Szőts M., Csirik J., Gergely T., Karvalics L.: MASZEKER: projekt szemantikus keresőtechnológia kidolgozására. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 159–167

2. Hussami P.: MASZEKER: szemantikus kereső program. In: Tanács A., Vincze V. (szerk.): MSzNy 2011 – VIII Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2011) 321–322
3. Hussami P.: MASZEKER: szemantikus kereső program. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 302–304
4. Zsibrita J., Vincze V.: Magyar nyelvű néprajzi keresőrendszer. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 361–367
5. Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W. (eds.): Handbook of Latent Semantic Analysis (Universita of Colorado Institute of Cognitive Science Series). Psychology Press (2007)
6. Ananiadou, S., Thompson, P., Nawaz, R.: Improving Search through Event-based Biomedicasl Text Mining. In: Darányi, S., Lendvai, P. (eds.): Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Comminication Texts (2010) 42–54
7. Abolhassani, H., K. S. Esmaili: A categorization scheme for semantic web search engines. In: 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06) (2006)
8. Nirenburg, S.: Homer, the author of the Iliad and the computational linguistic turn. In: Words and Intelligence II. Springer (2007)
9. Nirenburg, S., Raskin, V.: Ontological Semantics. The MIT Press (2004)
10. Dahlgren, K.: Technical overview of Cognition's semantic NLP (as applied to search). Technical report. Cognition Technologies, Inc. (2007) http://www.cognition.com/pdfs/Cognition_Semantic_NLP_for_Search_Overview.pdf.
11. Montalbano, E.: Microsoft testing Kumo search engine internally. NetworkWorld, March 3, 2009. WWW document. http://www.networkworld.com/news/2009/030309-microsoft-testing-kumo-search-engine.html (Letöltve: 2009. március 27.).
12. UpTake under the hood: the Interview. Alt-SearchEngines, 2008. május 14. WWW document. http://www.altsearchengines.com
13. Dietze, H., Schroeder. M.: GoWeb: A semantic search engine for the life science web. In: Burger, A., Paschke, A., Romano, A., Splendiani, A. (eds.): Proceedings of the Intl. Workshop Semantic Web Applications and Tools for the Life Sciences SWAT4LS. Edinburgh (2008)
14. Miyao, Y., Ohta, T., Masuda, K., Tsuruoka Y., Yosida K., Ninomiya T., Tsujii J.: Semantic Retrieval for the Accurate Identification of relational Concepts in Massive Textbases. In: Annual Meeting - Association for Computational Linguistics, Vol. 2 (2006) 1017–1024
15. Parsons, T.: Events in the Semantics of English: A Study in Subatomic Semantics, MIT Press, Cambridge, MA (1990)
16. A szemantikus nyelvészeti erőforrások áttekintése. http://www.maszeker.hu/?page=download
17. Fillmore, C. J.: Frame Semantics And The Nature Of Language. In: Annals of the New York Academy of Sciences, Vol. 280, No.1 (1976) 20–32
18. Baker, C. F., Fillmore, C. J., Lowe, J. B.: The Berkeley FrameNet Project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL '98), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA (1998) 86–90
19. Matsubayashi, Y., Okazaki, N., Tsujii, J.: A comparative study on generalization of semantic roles in FrameNet. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP (2009) 19–27

20. Shen, D., Lapata, M.: Using semantic roles to improve question answering. In: Proceedings of EMNLP-CoNLL (2007) 12–21
21. Kiss, M., Nagy, Á., Vincze, V., Almási, A., Alexin, Z., Csirik, J.: A Manually Annotated Corpus of Pharmaceutical Patents. In: Proceedings of TSD 2012 (2012) 135–142
22. Nakov, P., Hearst, M.: Search Engine Statistics Beyond the n-gram: Application to Noun Compound Bracketing. In: CoNLL-2005. Ann Arbor, MI (2005)

VIII. Poszterek és laptopos bemutatók

Dokumentumcsoportok automatikus kulcsszavazása és témakövetés

Ács Zsombor[1], Farkas Richárd[2]

[1] Szegedi Tudományegyetem
Acs.Zsombor@stud.u-szeged.hu
[2] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
rfarkas@inf.u-szeged.hu

Kivonat: A cikkben bemutatunk egy olyan algoritmust, mely a Látens Dirichlet Allokációt felhasználva, természetes nyelvű szöveghalmazt klaszterez, majd ezeket a csoportokat jól kifejező szavakkal felcímkézi. A kifejlesztett módszer alkalmas a dokumentumhalmaz időintervallumokra felosztott részein keletkezett klaszterhalmazok közötti összefüggések, átmenetek, illetve trendek feltárására. Kidolgoztunk egy olyan entrópiasúlyozáson alapuló címkézőt, mely empirikusan is jobb kulcsszavakkal látja el a klasztereket, mint a sztenderd módszerek (term-frekvencia, χ-négyzet statisztika).

1 Bevezetés

Az internet korának egyik jelentős tendenciája az adatok rohamosan növekvő mennyisége, melyek nagy része szöveges. A klaszterezés, illetve a csoportok felcímkézése egyre gyakrabban használt eszközzé válik a piackutatásokhoz, cikkbázisok, fórumok vagy blogok elemzéséhez, ill. a keresőmotorok informáltságának növeléséhez. Segítségével egy átfogó képet kapunk a dokumentumhalmaz szerkezetéről, ami a szöveges adat terjedelmét figyelembe véve, emberi erővel gyakorlatilag elképzelhetetlen lenne.

A klaszterező eljárások által meghatározott csoportok csak egy számítógépes reprezentációt adnak, mely a gyakorlatban az ember számára nem (vagy csak közvetett módon) értelmezhető, hiszen nem tudjuk, milyen dokumentumok és miért kerültek egy csoportba. Ezért szükséges a csoportokat a legrelevánsabb kulcsszavakkal, címkékkel ellátni.

Többféle megközelítés létezik a címkék meghatározására, legeredményesebbnek a differenciális csoportcímkéző eljárások [2] bizonyulnak. Erre a célra alkalmazhatók a vektortérmodell dimenziócsökkentéséhez is használt jellemzőkiválasztó módszerek. A munkánk során ilyen címkéző algoritmusokat vizsgáltunk, kidolgoztuk az egyik algoritmus kiterjesztését.

Végső célkitűzésünk az, hogy trendeket azonosítsunk, a témák időbeli lefolyását nyomon kövessük szöveges dokumentumok alapján. A trendkövetés egy információelméleti módszer, mely során különféle tendenciákat keresünk az elmúlt időszak adataiban, mellyel jóslást tehetünk a jövőre vonatkozóan. Ez egy meglehetősen új módszer, elsősorban a gazdasági életben – azon belüli is a pénzügyi szektorban – alkalmazzák. A kutatás során egy dátumokkal felcímkézett dokumentumgyűjteményt

osztottunk fel meghatározott időintervallumokra, és ezen „részkorpuszokon” automatikusan kialakított klaszterek közötti hasonlóságokat és tendenciákat figyeltünk meg.

2 Kapcsolódó munkák

Az alábbi fejezetben ismertetjük a Látens Dirichlet Allokáció (LDA) „előfutárát”, a Látens Szemantikus Indexelést, illetve magát a LDA-t. Ezen módszerek az eredeti jellemzők kombinálásából új, eddig nem létező jellemzőket generálnak, melyek száma kevesebb, mint az eredeti halmaz elemszáma, ezzel jelentős redukciót elérve. Először létrehozzák az új jellemzőket, majd a dokumentumokat az új reprezentációnak megfelelő alakra alakítják.

2.1 Látens szemantikus indexelés

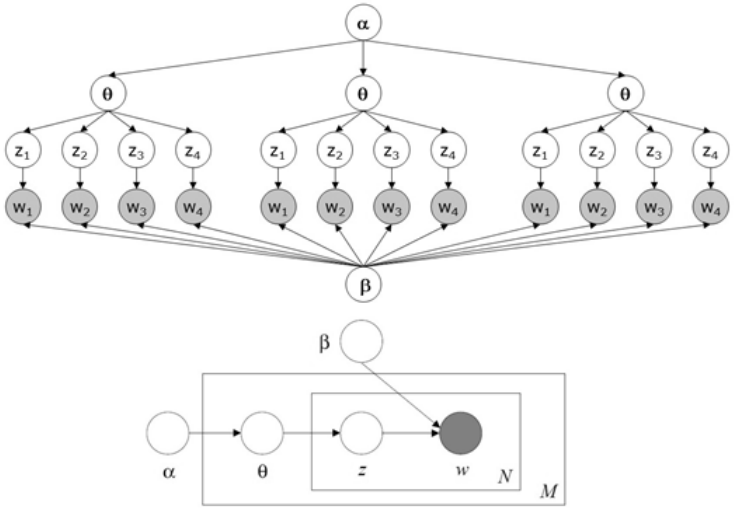
Az egyik legelterjedtebb módszer a látens szemantikus indexelés (LSI), mely egy szinguláris érték-felbontáson alapuló vektortér-transzformáció segítségével az eredeti dokumentumvektorokat kisebb dimenziójú vektorokká alakítja át, melyek meglepően jól jellemzik a korpusz rejtett szemantikai szerkezetét.

Az LSI egyik továbbfejlesztett változata a valószínűségi LSI (pLSI) [2]. A modell lényege, hogy minden dokumentumbeli szó felfogható egy kevert modell mintájának, melynek komponenseit témáknak hívjuk. Így egy dokumentum témák keverékeként értelmezhető.

A generatív valószínűségi modell egy olyan parametrikus modell, amely az egyes változók (paraméterek) olyan értékeit keresi (tanulja), amelyek legnagyobb valószínűséggel generálják a dokumentumkorpuszt. A pLSI hátránya, hogy nem ad generatív valószínűségi modellt a dokumentumok szintjén. Ebből adódóan, nincs természetes mód az előzőleg nem látott dokumentumok priori valószínűségének meghatározására a tanítóhalmazból. További hátrány, hogy a korpusz méretével lineárisan nő az optimalizálandó változók száma ($kV + kM$, ahol k a témák száma, V a szótár, M pedig a dokumentumhalmaz mérete).

2.2 Látens Dirichlet allokáció

A másik kisdimenziós témareprezentációt adó modell, a látens Dirichlet allokáció, többek között a pLSI hátrányait hivatott „kijavítani”. Az LDA is egy generatív valószínűségi modell, amely egy korpusz dokumentumait reprezentálja rögzített számú téma keverékeként. A témákat a szótár felett vett multinomiális valószínűségeloszlásokkal reprezentálja, egy dokumentum pedig ezen eloszlások keveréke (explicit reprezentációt adva) [1]. Így a rejtett multinomiális változók maguk a témák. A modell paramétereinek száma $k + kV$, azaz nem nő a korpusz méretével (ellentétben a pLSI-vel). Az LDA egy háromszintű hierarchikus Bayes-modell, mely a 1. ábrán (fent) látható.



1. ábra. Az LDA Bayes-hálója (fent), illetve gráfikus modellje (lent) [1].

Az LDA gráfikus modelljének reprezentációja az alsó ábrán látható. A téglalapok felfoghatók egymáson lévő lemezeknek, melyek halmazokat ábrázolnak. A külső téglalap a dokumentumok, míg a belső a dokumentumon belüli témák, illetve szavak reprezentációja. α és β hiperparaméterek, az uniform Dirichlet eloszlás paraméterei, előbbi a dokumentumszintű témaeloszlások, utóbbi pedig a szó-téma eloszlás felett. θ_i a témaeloszlást adja meg dokumentumszinten, z_{ij} az i-edik dokumentum j-edik szavához tartozó témát jelenti, w_{ij} pedig az adott szót. Az egyetlen megfigyelhető változó a w_{ij}, a többi rejtett változó.

3 Módszer

A kutatásokhoz rendelkezésre állt az *origo.hu* hírportál *Techbázis* rovatának több mint 10 éves archívuma (1998-2009), mely megközelítőleg 20 000 dokumentumot tartalmaz. A kifejlesztett módszer azonban alkalmas tetszőleges korpusz feldolgozására.

3.1 Előfeldolgozás

A kutatás során reprezentációs modellként a klasszikus vektortérmodellt (VTM) alkalmaztuk.

Előfeldolgozási lépésként a dokumentumhalmazt féléves partíciókra bontottuk, majd elvégeztük a nyers szövegek tokenizálását, lemmatizálását, illetve stopszószűrését [3]. A korpusz szótövezése, illetve a stopszavak eliminálása kulcsfontosságú, általuk jelentősen csökken a VTM dimenziószáma, mely felgyorsítja az LDA

futását, csökkenti a szükséges tárhely méretét, másrészről pedig nagymértékben javít az LDA által generált reprezentáció minőségén, hatékonyságán.

3.1 Kulcsszórangsor

Az LDA modellben is létezik a témákhoz tartozó szavaknak egy relevancia-sorrendje, minden témabeli szó egy súllyal szerepel a témában. Ezen súly meghatározása a dokumentumreprezentálásban általában alkalmazott *term-frekvencia* (TF) súlyozáson alapul – de a dokumentum helyett, a témák szintjén értelmezendő. Így az LDA a következő képlettel számolja a *k*-adik lexikonbeli szó *i*-edik témabeli gyakoriságát:

$$f_{ki} = \frac{n_{ki}}{\sum_{k=1}^{K} n_{ki}}$$

ahol n_{ki}, a *k*-adik lexikonbeli term *i*-edik témabeli előfordulásainak száma (mely a dokumentumok feletti eloszlásból adódik), *K* a lexikon mérete.

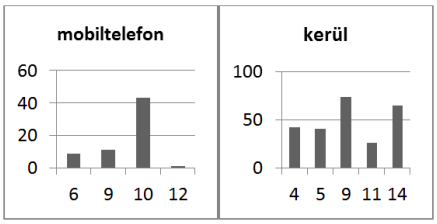
A *χ-négyzet statisztika* olyan klaszterező eljárások utáni címkézésre alkalmas, amelyek egy dokumentumot csak egy csoportba sorolhatnak be. Ezzel szemben az LDA egy valószínűség-eloszlást ad meg dokumentumonként a témák felett.

Lehetséges megoldás lehet, ha egyszerűen vesszük a legnagyobb valószínűséggel rendelkező témát, és azt rendeljük a dokumentumhoz. Így már alkalmazható az eredeti koncepció. Szofisztikáltabb megközelítés, ha megtartjuk a dokumentumok feletti eloszlást, és ennek megfelelően módosítjuk a χ-négyzet statisztika számítási módját:

$$\chi^2(t_i, c_j) = \frac{M \cdot \left(n_{t_i c_j} n_{\overline{t_i c_j}} - n_{t_i \overline{c_j}} n_{\overline{t_i} c_j}\right)^2}{n_{c_j} \cdot n_{\overline{c_j}} \cdot n_{t_i} \cdot n_{\overline{t_i}}}$$

ahol M a dokumentumok számát, t_i az *i*-edik lexikonbeli termet, c_j pedig a *j*-edik témát jelenti. n_{c_j} a dokumentumok *j*-edik klaszterhez tartozásának valószínűségösszege: $\sum_{k=0}^{M} p(d_k|c_j)$. Ebből következik, hogy $n_{\overline{c_j}} = N - n_{c_j}$. n_{t_i} az *i*-edik termet tartalmazó dokumentumok száma, azaz $n_{\overline{t_i}} = N - n_{t_i}$. $n_{t_i c_j}$ az *i*-edik termet tartalmazó dokumentumok *j*-edik klaszterhez tartozásának valószínűségösszege: $\sum_{k=0}^{M} p(d_{k,t_j}|c_j)$. A nem említett mennyiségek a fentiekhez hasonló módon számíthatók ki.

Az általunk kidolgozott kulcsszórangsoroló módszer matematikai alapja az entrópia, mely egy rendszer rendezetlenségi fokát méri. Az alábbi grafikonok a *TechBázis* korpusz 2007/1 félévében levő témaeloszlásokat ábrázolják, a *mobiltelefon*, illetve a *kerül* termekre. Érezhető, hogy a *mobiltelefon* szó sokkal jobban jellemezhetne egy témát, mint a *kerül*.



2. ábra. Termek eloszlása a témák felett.

A címkézés szempontjából releváns szavak entrópiája alacsony, azaz jellemzően egy kimagasló témával rendelkező grafikonnal ábrázolhatók. Csupán entrópiával nem lehet a témákat felcímkézni, hiszen ez az érték egy adott termre minden témában azonos. Az alacsony entrópiájú szavak vélhetően kevés témában vannak jelen nagy számmal, így kihasználhatjuk a termek témabeli gyakoriságát is. Azaz egy alacsony entrópiával rendelkező term azokban a témákban, melyekben nagy előfordulási számmal rendelkezik, hatványozottan magas értéket fog kapni. Lehetséges megoldás, ha vesszük a TF és entrópia egy kombinációját. A *k*-adik lexikonbeli szó *i*-edik témabeli TF-entrópia mértéke a következő:

$$tfent_{ki} = f_{ki} \cdot ((uniform - H(t_k)) + 1)^{\alpha}$$
$$= \frac{n_{ki}}{\sum_{k=1}^{K} n_{ki}} \cdot \left(uniform + \sum_{i=0}^{N} \frac{n_{ki}}{\sum_{j=0}^{N} n_{kj}} \cdot \ln \frac{n_{ki}}{\sum_{j=0}^{N} n_{kj}} + 1 \right)^{\alpha}$$

ahol *K* a szótár mérete, *N* a témák száma, α pedig az entrópia súlyozásának paramétere, mellyel beállíthatjuk az optimális entrópiaarányt. Az *uniform* egy konstans, az előforduló összes entrópia közül a maximumot jelenti. Mivel az alacsony entrópia érték jelent magasabb relevanciát a számunkra, negatív előjellel szerepeltetjük azt. A hozzáadott 1 csupán a [1, uniform+1] értéktartományba való transzformálást eredményezi, hogy az α paraméter hatása minden esetben az entrópiaarány növelését eredményezze.

3.2 Klasztermegfeleltetés

A féléves periódusok klaszterei közötti kapcsolatok feltárására alapvetően három módszert használtunk: halmazmetszeten, vektortávolságon alapuló számításokat, illetve az operációkutatásból ismert hozzárendelési feladatot. A trendek, illetve tendenciák meghatározása egyben a címkéző algoritmusok kiértékelése is volt. A munkahipotézisünk az volt, hogy egy címkéző akkor tekinthető eredményesebbnek, ha az általa felcímkézett klaszterek jobban képezhetők le egy másik, következő félév klaszтereire, azaz erősebb megfeleltetések találhatók a két félév között.

A halmazmetszeten alapuló megközelítés során a témákhoz rendelt szavak címkesúly szerint csökkenő sorrendbe rendezett listájából vettünk egy felső részt (meghatározott százalékot vagy darabszámot), és ezeket a részhalmazokat hasonlítottuk össze.

Vektortávolság esetén a témákat nem csupán a hozzá tartozó szavak halmazaként fogtuk fel, hanem minden term egy súlyértékkel szerepel. Ekkor minden téma felfogható egy vektornak a lexikon szavai által kifeszített vektortérben. Előnye, hogy sokkal jobban reprezentálja az adott témát, hiszen folytonos értékeket használunk. A témavektorok távolságának meghatározását két távolságfüggvény segítségével végeztük: euklideszi, illetve Manhattan-távolság.

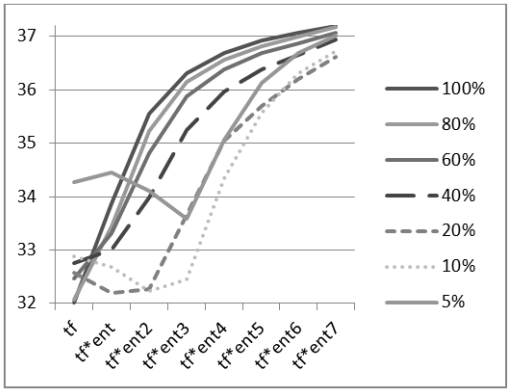
Mindkét esetben a klaszterösszerendelési mátrixon végeztünk maximum (illetve minimum) keresést. A módszer hátránya, hogy egy közel uniform eloszlású sor maximális eleme nem feltétlenül jelent egy valódi átmenetet. Ezen sorok általában a címkéző „gyengeségét" hivatottak jelezni, hiszen közelebb hozzák egymáshoz a klasztereket, így nehezebb a köztük lévő átmenetek felismerése. Az ilyen mátrixok (illetve sorok) „büntetését" a *sorentrópiák* számolása oldotta meg. Ha összegezzük a sorok entrópiáját, kaphatunk egy „jósági értéket" a címkézőre vonatkozóan, melyet a továbbiakban nevezzünk *mátrixentrópiá*nak. Annál jobb a kulcsszórangsor, minél alacsonyabb a mátrixentrópia.

Az entrópiaalapú kiértékelés nem vette figyelembe a metszethalmazok nagyságrendjét, ez motiválta a hozzárendelési feladat alkalmazását. A klaszterek felfoghatók egy gráf csúcsainak, a köztük lévő összehasonlítási mátrix pedig az élek súlyait reprezentálja. Azaz egy teljes páros gráfról beszélünk, ahol mindkét partíció minden csúcsára fennáll, hogy vezet belőle él a másik partíció minden csúcsába. A cél, hogy ebben a gráfban keressünk egy maximális (vektortávolság esetén minimális) összsúlyú teljes párosítást. Így globálisan, a teljes mátrixban találhatunk egy optimális, egy az egyhez összerendelést. A végső jósági érték – azaz a kulcsszórangsorolás minősítése – pedig ezen algoritmus által választott párosítások súlyösszege.

4 Eredmények

Az *entrópiasúlyozás* eredményeit összehasonlítottuk az LDA által adott alap kulcsszó rangsorral (*term-frekvencia*), illetve a hagyományos χ*-négyzet statisztikával*. Emberi kiértékelés alapján a legjobb eredményt produkálta az általunk fejlesztett algoritmus, azaz kifejezőbb címkékkel látta el, mint az alapmódszerek.

A címkéző módszereket automatikus, objektív kiértékelés alá is vetettük, és részben sikerült empirikusan bebizonyítani a saját címkéző jó szereplését. A következő ábra (jobb oldal) a mátrixentrópiák alakulását mutatja különböző címkéző módszerek esetén. Az értékeket a $[0, max - min]$ tartományba transzformáltuk a kifejezőerő növelése érdekében. Az ábra bal oldalán az *entrópiasúlyozás* eredményei láthatóak a növekvő α paraméterének függvényében.



	tf	ent^3	$\chi\,(sz)$	$\chi\,(lt)$
100%	32,022	36,318	38,659	38,881
80%	32,073	36,138	38,657	38,880
60%	32,474	35,881	38,546	38,835
40%	32,755	35,253	38,213	38,594
20%	32,583	33,650	37,989	38,576
10%	32,874	32,446	38,201	38,482
5%	34,265	33,588	38,280	38,438

3. ábra. Mátrixentrópiák alakulása különböző címkéző módszerek esetén.

Megállapítható, hogy kis felső rész vétele esetén *(5%, 10%)* az entrópia kismértékű bevonása (ent^2, ent^3) által csökkent a mátrixentrópia, azaz ekkor könnyebb a klasztermegfeleltetések meghatározása. Ez részben tekinthető egy sikeres bizonyításnak, hiszen a címkézők gyakorlati alkalmazása általában kevés számú címke használatára korlátozódik, és ezen a kisméretű felső részen az entrópiasúlyozás szerepelt a legjobban.

A hozzárendelési feladattal történő kiértékelés során hasonló eredményre jutottunk.

5 További munkák, diszkusszió

A cikkben csupán az egy-egy klasztermegfeleltetéssel foglalkozunk, azaz azzal a feltételezéssel éltünk, hogy a témák halmaza időben lassan változik, egy téma kihalása vagy születése igen ritka. A vizsgált *Techbázis* rovat alapján az esetek nagy részében valóban egy-egy megfeleltetés volt, azonban előfordul a klaszterszétválás, illetve összeolvadás is. A javasolt modellbe bevonhatóak a klaszterek *születésének* és *halálának* esetei is.

A rendszer fő erőssége, hogy segítségével rövid idő alatt határozhatunk meg egy nyers dokumentumhalmazon témákat, melyek egy átfogó, jól használható képet adnak a korpusz tartalmi elemeiről. A trendkövetés nagy előnye, hogy jóslást tehetünk a jövőbeli témákra vonatkozóan, mellyel előre jelezhetők akár bizonyos piacmozgási folyamatok is.

Köszönetnyilvánítás

Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

1. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Journal of Machine Learning Research, Vol. 3, No. 5 (2003) 993-1022
2. Tikk D. (szerk.): Szövegbányászat. TypoTEX, Budapest (2007)
3. Zsibrita J., Vincze V., Farkas R.: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 275–283

Egy hatékonyabb webes sablonszűrő algoritmus – avagy miként lehet a cumisüveg potenciális veszélyforrás Obamára nézve

Endrédy István[1], Novák Attila[1]

[1] MTA–PPKE Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter utca 50/a
{endredy.istvan.gergely, novak.attila}@itk.ppke.hu

Kivonat: A folyamatosan növekvő web[1] tartalmából hatékonyan lehet nagyméretű korpuszt építeni. Azonban a dinamikusan generált weblapok gyakran sok irreleváns és ismétlődő szöveget tartalmaznak, amelyek egyes sablonos szövegrészeket, kifejezéseket felülreprezentálva rontják a korpusz minőségét. Ebben a cikkben olyan automatikus szövegkinyerő eljárást mutatunk be, amely a korábbi módszereknél hatékonyabban minimalizálja az irreleváns ismétlődő részek előfordulását a webről letöltött korpuszokban.

1 A feladat

A webről nyert korpuszok építésekor az általában dinamikusan generált HTML-tartalomból történő szövegkinyerés nem triviális feladat a különböző oldalakon ismétlődő rengeteg irreleváns sablonos tartalom miatt. A feladatot az angol irodalomban boilerplate removalnek, sabloneltávolításnak nevezik. A művelet lényege, hogy a HTML tartalomból csak az értékes részeket igyekszik kiszűrni, a menük, fej- és láblécek, reklámok, a minden oldalon ismétlődő struktúra kiszűrésével.

A fenti feladatra számos algoritmus létezik. Ezek közül két szabadon hozzáférhető implementációval rendelkező megoldást alaposabban is szemügyre vettünk.

1.1 A Body Text Extraction (BTE) algoritmus

A BTE [1] a weblap azon általában jellemző tulajdonságát használja ki, hogy a boilerplate tartalomban sűrűbben szerepelnek a HTML-tagek. Ezért megkeresi azt a leghosszabb részt, ahol a tagek száma minimális. Az ötlet egyszerű, azonban sokszor téved, és nem képes szöveget kinyerni olyan helyzetekben, ahol az értékes részben a feltételezéssel ellentétben mégis sűrűbben fordulnak elő HTML-tagek, például ha táblázat szerepel benne, vagy egy reklámokkal szabdalt cikk esetében. Ilyenkor az értékes tartalom jelentős része (akár az egész) elveszhet, esetleg helyette teljes egészében irreleváns tartalom jelenik meg.

[1] 50 milliárd oldal, http://www.worldwidewebsize.com (2012.09.20.)

1.2 JusText algoritmus

A JusText algoritmus [2] a HTML tartalmat bekezdésekre bontja a szöveget tartalmazó tagek mentén. Minden egységben megszámolja a benne szereplő linkek, stopwordök és szavak számát. Ezek alapján osztályozza őket: bizonyos küszöb mentén *jó*, *majdnem jó* illetve *rossz* kategóriákba. Majd a *jókkal* körülvett *majdnem jók* szintén bekerülnek a *jók* közé, és így születik meg az értékes szöveg: a *jó* bekezdések. Az algoritmus elég jó minőséget ad még extrém oldalakon is.

Az algoritmus használatával 2012 nyarán magyar hírportálokról kinyert tartalmat vizsgálva azonban feltűnt, hogy a kapott korpusz még mindig a várthoz képest nagyon erősen túlreprezentált kifejezéseket tartalmaz, mint az alábbi példák mutatják:

(1) Utasi Árpi-szerű mesemondó . (10587 db)
cumisüveg potenciális veszélyforrás . (1578 db)
Obama amerikai elnök , (292 db)

(2) etióp atléta: cseh jobbhátvéd (39328 db)
Barack Obama amerikai elnök (2372 db)
Matolcsy György nemzetgazdasági miniszter (1633 db)
George Bush amerikai elnök (1626 db)

Azt találtuk, hogy a problémát elsősorban egyrészt a cikkek alján található (kapcsolódó) cikkajánlók nem megfelelő kiszűrése, másrészt a kizárólag ilyeneket tartalmazó oldalak okozzák.

2 Az aranyásó algoritmus

A problémát az adott egyedi weboldalaknál magasabb szintre lépve sikerült a korábbinál hatékonyabban megoldani. Eleinte abba az irányba indultunk, hogy az egyes weblapok nem kívánt részeit távolítsuk el. Mikor nem sikerült átütő eredményt elérni, akkor jutott eszünkbe, hogy miért a rosszat keressük, miért nem az értékes részeket? Ahogy az életben is érdemes a jót keresni, így tettünk a weboldalak esetén is: így született meg az aranyásó algoritmus.

A megoldás azon alapul, hogy egy adott webdoménen/aldoménen belül a dinamikusan generált weboldalak, illetve url-ek nem szöveges tartalma (pl. a HTML-kód) általában tartalmaz közös mintázatokat, amelyeket azonosítva megtalálható az értékes tartalom. Az algoritmus az adott domén oldalaiból mintát vesz, és megpróbálja megkeresni azt a HTML-taget, amelyen belül (az oldalak zömében) a cikk található, különös tekintettel azokra a mintákra, amelyek az oldalakon ismétlődnek. Például gyakori a hírportálokon, hogy a cikk alján feltüntetik a legnépszerűbb 5 cikk ajánlóját. Ezeket a szövegeket a korábbi sablonszűrő algoritmusok nem szűrik ki (pl. a justext a cikk részének veszi őket), mert önmagukban nem rossz szövegek, viszont duplikátumot okoznak majd a korpuszban, erősen felülreprezentálva az ezekben található szöveget.

Minden doménre megtanuljuk azt a HTML szülő taget, amely csak a cikket tartalmazza, majd csak ezen tag tartalmát adjuk oda a sablonszűrő algoritmusnak. Előnye, hogy azon oldalak, ahol nincs cikk (tematikus nyitólapok, címkefelhők, keresőlap-eredmények, stb.), ott az algoritmus nem ad semmit, hiszen a doménre jellemző cikk tag nincs jelen. Így az algoritmus automatikusan kiszűri ezeknek a lapoknak a tartalmát, ami örvendetes. Azokon a lapokon pedig, ahol valóban van cikk, a sablonszűrő algoritmus már csak a lényegi tartalmat kapja, erősen megkönnyítve a dolgát.

2.1 Az algoritmus részletes leírása

Első lépésként megtanuljuk a doménre jellemző HTML-mintázatot, ez a tanulófázis. Az algoritmus letölt pár 100 oldalt, és lefuttatja rajtuk a sablonszűrő algoritmusokat (justext), amely minden oldal tartalmát bekezdésekre bontja és értékeli. Az Aranyásó algoritmus az egyes kinyert bekezdéseket átnézi: ismétlődnek-e más oldalakon a letöltött mintában. Ha igen, akkor ezeket rossz bekezdésnek minősíti, majd megkeresi a jó bekezdések legközelebbi, közös szülő tagjét a DOM hierarchiában. A tanulófázis végén a leggyakoribb jó szülő tag lesz a győztes. Nem kapunk optimális eredményt azonban, ha végpontként ennek a tagnek a zárópontját választjuk. Előfordul ugyanis, hogy a teljes cikket tartalmazó szülő tag nem kívánt bekezdéseket is tartalmaz. Ilyen esetben az algoritmus nem találná meg az optimális vágási pontokat. Ezért az első körben választott tartalmon belül újabb keresést végzünk az optimális végpontra, amely több tagből álló sorozat is lehet. Miután ezt is kiválasztottuk, az algoritmus (természetesen a tanulófázis oldalaiból is) csak az így kivágott rész tartalmát adja át a sablonszűrőnek.

Az Aranyásó csak azokból az oldalakból tanul, amelyben a kinyert bekezdések hossza elér egy minimumot: pár doménen, ahol rendkívül gyakoriak a tényleges tartalmat nem adó gyűjtőlapok, e nélkül nem sikerült a megtanulni a legjobb vágási pontot.

3 Eredmények

Az Aranyásó algoritmussal jelentősen sikerült csökkentenünk a legyűjtött korpuszban szereplő ismétlődéseket. Három doménen (origo.hu, index.hu, nol.hu) való futtatással 2000 oldal letöltése után a bejárást leállítva kapott eredményeinket a következő oldalon szereplő táblázatban foglaljuk össze.

1. táblázat: Sablonszűrő algoritmusok összehasonlítása az egyes doméneken.

Algoritmus		Kinyert mondatok száma	Egyedi mondatok száma	Össz karakter-szám	Egyedi monda-tok karakter-száma	Egyedi mondatok aránya %	Egyedi mondatok aránya karakter-számban %
origo	összes szöveg	264 423	63 594	16 218 753	7 048 011	24%	43%
	BTE	60 682	33 269	12 016 560	7 499 307	54%	62%
	JusText	58 670	30 168	8 425 059	4 901 528	51%	58%
	aranyásó	22 475	21 242	3 076 288	3 051 376	94%	99%
nol.hu	összes szöveg	509 408	144 003	25 358 477	12 570 527	28%	49%
	BTE	154 547	107 573	24 292 755	13 544 130	69%	55%
	JusText	186 727	128 782	14 167 718	11 665 284	68%	82%
	aranyásó	162 674	123 716	12 326 113	11 078 914	76%	89%
index.hu	összes szöveg	232 132	55 466	9 115 415	4 542 925	23%	49%
	BTE	51 713	26 176	5 756 176	4 061 697	50%	70%
	JusText	40 970	29 223	4 371 693	3 441 337	71%	78%
	aranyásó	13 062	11 887	1 533 957	1 489 131	91%	97%

Az eredmények világosan mutatják, hogy az algoritmus hatékonyan csökkenti a kinyert korpuszban szereplő felesleges ismétlődéseket. Arra vonatkozó becslést egyelőre nem végeztünk, hogy ugyanakkor mennyi értékes anyagot veszítünk esetleg el, és hogy ebből a szempontból az eredmény hogyan viszonyul az egyéb algoritmusok teljesítményéhez.

Nézzük tehát, hogy a módosított algoritmust novemberben futtatva hogyan teljesít Obama elnök a cumisüveggel szemben (hasonlóan az (1) példához, a kifejezés után álló írásjel itt is a minta részét képezi):

(3) Obama amerikai elnök , (47 db)
Utasi Árpi-szerű mesemondó . (1 db)
cumisüveg potenciális veszélyforrás . (1 db)

Látjuk, hogy az eredeti mintában a vesszőt is tartalmazó '*Obama amerikai elnök ,*' kifejezés maga is felül volt reprezentálva, nem is beszélve a másik kettőről. Mi történt ugyanakkor az etióp atlétával és a cseh jobbhátvéddel?

(4) Matolcsy György nemzetgazdasági miniszter (694 db)
Barack Obama amerikai elnök (664 db)
Sólyom László köztársasági elnök (367 db)

Angela Merkel német kancellár (345 db)
...
etióp atléta: cseh jobbhátvéd (1 db)

Eltűntek ők is ki kicsoda-listánkról. Bár a teljes domént az algoritmus a cikk írásakor még nem járta be, ezért adataink nem teljesen összehasonlíthatók a korábbiakkal, megnyugtató, hogy a gyanús kifejezések eltűntek a gyakori kifejezések listájáról.

Hivatkozások

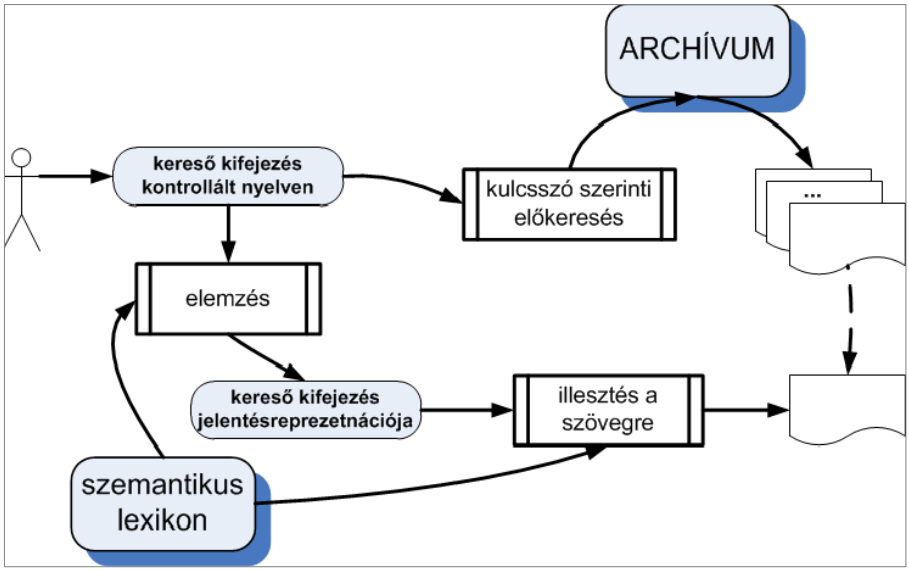
1. Finn, A., Kushmerick, N., Smyth, B.: Fact or fiction: Content classification for digital libraries. In: DELOS Workshop: Personalisation and Recommender, Systems in Digital Libraries (2001)
2. Pomikalek, J.: Removing Boilerplate and Duplicate Content from Web Corpora. Masaryk University Faculty of Informatics, Brno, 2011. http://code.google.com/p/justext/ (2012. május 20.)

MASZEKER: szemantikus kereső program

Hussami Péter[1]

[1]Alkalmazott Logikai Laboratórium
1022 Budapest, Hankóczy j. u. 7
hussami@all.hu

Az Alkalmazott Logikai Laboratórium és a Szegedi Tudományegyetem Informatikai Tanszékcsoportja, valamint Könyvtár- és Humán Információtudományi Tanszéke egy, a Nemzeti Fejlesztési Ügynökség által támogatott közös projektet (TECH_08_A2/2-2008-0092) fejezett be 2012-ben. A projekt célja olyan, új elveken alapuló, integrált információ-visszakereső rendszer kifejlesztése volt, amely a keresést végző felhasználó szemantikai kompetenciáját az eddigieknél nagyobb mértékben kiaknázva teszi lehetővé a természetes nyelvi dokumentumtárakban (szövegekben) történő valóban *tartalmi* keresést. Egyszerűen szólva: a felhasználó jól formált frázisokkal, mondatokkal specifikálhatja, milyen tartalmú dokumentumokat keres. A rendszer áttekintő architektúrája az 1. ábrán látható.



1. ábra. A MASZEKER rendszer áttekintő architektúrája.

Az ábrának megfelelően a releváns dokumentumok keresése a következő lépésekből áll:

1.) a felhasználó egy kontrollált nyelven adja meg a keresőkifejezést,

2.) a rendszer szintaktikus és szemantikus elemzést végezve előállítja a keresőkifejezés jelentésreprezentációját,
3.) szavak szerinti kereséssel előszűri az archívumot,
4.) végül azokra a szövegszegmensekre, amelyekben a szavak szerinti keresés találatai vannak, illeszti a keresőkifejezés jelentésreprezentációját.

Az MSzNy VII konferencián tartott előadáson [1] ismertetésre kerültek a fenti elemek megvalósítására vonatkozó elméleti alapelvek, elsősorban a szemantikus reprezentáció felépítése mint sarokkő köré szervezve. Az MSzNy VIII konferencián tartott bemutatón [2] a rendszer első változatát mutattuk be, amely főnévi csoportokon mint keresőkifejezéseken működött. Idén a teljes, jól formált főnévi csoportokon és mondatokon működő rendszert kívánjuk bemutatni. Ugyanezen a konferencián egy előadás [3] számol be a rendszer alapjául szolgáló technológiáról.

A demóban az archívumot szabadalmi leírások főigénypontjaiból összeállított dokumentumgyűjtemény alkotja[1]. A keresőkifejezés több mondatból, ill. főnévi kifejezésből állhat, kontrollált angol nyelven megfogalmazva. A megszorítások az egyértelműséget biztosítják, a tipikusan nehezen egyértelműsíthető fordulatokat akartuk kizárni. Legfontosabb korlátozások (a teljes definíció [4]-ben hozzáférhető):

- csak kijelentő módú, jelen idejű mondatok használhatók,
- tiltott a mellérendelő mellékmondat (viszont a mondatok AND, OR kapcsolóval kapcsolhatóak, zárójelezhetőek),
- tiltott az alárendelő mellékmondatok bármiféle lerövidítése (pl. igeneves utómódosítók),
- az alárendelő mellékmondatnak a „which” vonatkozó névmással kell kezdődnie, és ennek a közvetlenül megelőző főnévi csoport fejére kell vonatkoznia,
- tiltottak az igeneves előmódosítók,
- felsorolás, koordináció csak főnévi csoportok közt megengedett, ezeket a felhasználónak jelölnie kell.

A felhasználói interfész segíti, és a morfoszintaktikai elemzés eredménye alapján ellenőrzi a szabályok betartását. Mivel a teljes szabályrendszer nem ellenőrizhető, a generált jelentésreprezentáció grafikusan bemutattatik – ha szükséges, a felhasználó módosíthatja a keresőkifejezést. Ez a megjelenítés segíti egyértelműsíteni az egyes szavak jelentésének megállapítását is, mivel ha több frame/synset van egy csomóponthoz rendelve, akkor a felhasználó választhatja a megfelelőt.

A rendszer a keresőkifejezéshez illő frázisokat keres az igénypontok szövegében, és az eredményt a grafikus interfészen megmutatja, kiemelve azokat a szavakat, amelyek olyan frázist alkotnak, melyet a keresőkifejezés egy szegmenséhez hasonlónak talált. Míg a keresőkifejezés feldolgozásánál maximálisan törekszünk a pontos jelentésreprezentációra, a keresés fázisában az aktuális szövegrészlet vizsgálatánál csak azt ellenőrizzük, hogy jelentheti-e a keresőkifejezés valamely frázisát.

[1] A projekt egyik kiemelt felhasználási területe a szabadalmi keresés, s a prototípust „gyógyhatású készítmények és kozmetikai szerek” témaköréből származó szabadalmakon mutatjuk be.

Köszönetnyilvánítás

A fejlesztés az NFÜ által finanszírozott, MASZEKER kódnevű, TECH_08_A2/2-2008-0092 számú projekt keretében valósult meg.

Hivatkozások

1. Szőts M., Csirik J., Gergely T., Karvalics L.: MASZEKER: projekt szemantikus kereső technológia kidolgozására In_ Tanács A., Vincze V. (szerk.): MSzNy 2010 – VII Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 159–167
2. Hussami P.: MASZEKER: szemantikus kereső program In: Tanács A., Vincze V. (szerk.): MSzNy 2011 – VIII Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2011) 321–322
3. Szőts M., Simonyi A.: Frame-szemantikára alapozott információ-visszakereső rendszer In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 275–285
4. A kontrollált nyelv definíciója http://www.maszeker.hu/?page=download

PureToken: egy új tokenizáló eszköz

Indig Balázs[1]

[1]Pázmány Péter Katolikus Egyetem, Információs Technológiai Kar,
MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter u. 50/a
indba@digitus.itk.ppke.hu

Kivonat A szövegek mondatra és tokenekre bontása manapság már nem aktív terület, így a rendelkezésre álló eszközök, amelyek ezt a feladatot végzik, a karbantartás és fejlesztés hiányától szenvednek. A jelenleg rendelkezésre álló, mondatra és tokenre bontó legjobb magyar eszköz, a Huntoken fejlesztése régóta nem aktív, viszont számtalan projektben van szükség egy ilyen eszközre. A Huntoken készítésekor a kornak megfelelő technológiákat alkalmaztak a szerzők, mint például a Latin-2 karakterkódolás és a Flex lexikaielemző-generátor. Ezek a technológiák mára elavultak, és átvette a helyüket más, például a Unicode-karakterkódolás. A jelen tanulmányban bemutatunk egy eszközt, amely a Huntoken alapjaiból kiindulva és a részletes specifikációs teszteket felhasználva, azzal teljesen azonos kimenetet képes generálni, ám Unicode alapon. Bemutatunk egy olyan változatot is, amely egy beépített morfológiai elemzőt (a Humort) felhasználva kisebb méretűvé válik, viszont egy átláthatóbb megoldáson alapul. Ez a rendszer – bár nyelvfüggő, de – a más nyelvekre való jó minőségű kiterjeszthetőség lehetőségét is magában hordozza. Rövid távú célunk, hogy a létrehozott új eszköz sebességében és a kimenet minőségében is megegyezzen, sőt hogy meghaladja a Huntokent. Az első mérések egy kimondottan erre a célra összeállított korpusszal készülnek.

1. Bevezetés

2003-ban az első Magyar Számítógépes Nyelvészeti Konferencián bemutatták a Huntoken szabályalapú mondatra bontót és tokenizálót [1]. Az akkori mérésekből kiderült, hogy a szabályalapú módszer sokkal hatékonyabb, mint a statisztikai gépi tanuláson alapuló változatok. Akkor a program 99,03%-os hatékonyságot ért el a Szeged-korpuszon [4]. Ezzel a problémakört megoldottnak tekintette mindenki, és a program fejlesztése abbamaradt. Az azóta eltelt majdnem 10 évben az informatika és a természetesnyelv-feldolgozás is változott, a Huntoken mára majdnem minden nyelvtechnológiai alkalmazás előfeldolgozó programjává vált, pusztán a hatékonysága miatt, ezáltal egy nélkülözhetetlen eszközzé nőtte ki magát.

Eközben az informatikai fejlődés sok szempontból elavulttá tette és az így jelentkező problémák áthidalása egyre több munkát okozott mindenkinek. Eljött

az idő, hogy aktualizálják a programot, de úgy, hogy az a jelenlegi hatékonyságát is tartsa meg. A program számtalan apró technikai változtatáson esett át és néhány újabb időközben megjelent mintát is kapott. Ezek rövid bemutatásra kerülnek a következő fejezetben.

2. Változások a Huntokenhez képest

A Huntoken alapját a GNU Flex lexikaielemző-generátor adta, amely a programozási nyelvek területén egy nagy múltra visszatekintő eszköz. Az elsődleges felhasználási területe a programozási nyelvek, melyek az angol nyelvből merítenek ihletet, így a világban bekövetkező nemzetközi trendekre érzéketlen. Ez az oka, hogy nem támogatja az Unicode karakterkódolást, ami majd 10 év alatt de facto internetes szabvánnyá vált a Latin-karaktertáblák helyett. Erre irányuló fejlesztési törekvések nincsenek is tervben a kicsiny igények miatt. Ez már a Huntoken használatánál is plusz munkát eredményezett[1]. A Flex alapot le kellett cserélni egy másik, hasonló programra. A választás a Quex nevű lexikaielemző-generátorra [3] esett, aminek elsődleges célkitűzése az Unicode támogatása a lexikális elemzésben. A program aktív fejlesztésnek örvend, sokan használják a tudása és a gyorsasága miatt, így kiváló új alapot teremt a Huntoken átiratának. A Quex Python-alapú elemzővel C vagy C++ forráskódot képes generálni, a Flexéhez nagyon hasonló felépítésű fájlokból. És ezzel teljesen platformfüggetlen ellentétben a GNU Flex-szel.

Az egyes szűrők szükség szerint át lettek csoportosítva a következőképpen:

- a *latin1* és *clean* szűrők összevonásra kerültek a *clean* szűrőbe
- a *abbrev* és a *abbrev_en* szűrők összevonásra kerültek
- a *sentbreak* szűrő törlésre került
- az *abbrev* szűrőből kikerültek a nem oda való korrekciós minták
- a token szűrő szétszedésre került több logikai részre.

Az egyes szűrők működésében is felléptek változások. A *clean* szűrő a lehető legtöbb entitást felismeri és visszaalakítja a Unicode megfelelőjére. Bemutatásra került egy új szűrő, az *escape*, amely azért felelős, hogy a mezőelválasztó karaktereket levédje olyan módon, hogy lehetőség szerint HTML-entitássá alakítsa[2].

Az *abbrev* szűrő felhasználta az M4 makrógenerátort, illetve egy Bash scriptet a rövidítésfájlok feldolgozására és a rövidítések a Flex fájlba, mintaként történő beillesztésére. Ennek a mechanizmusnak több hibája is volt, amelyek javításra kerültek. Ezeket röviden ismertetem:

- A rövidítésfájlok duplikációkat tartalmaztak. Ezek kétszer kerültek be a belőlük generált mintába.
- A rövidítéseket tartalmazó minta végére lezáróelemként a „nyug.” rövidítést mindenképpen beillesztette.

[1] Lásd clean és latin1 szűrők.
[2] Alapesetben ez a kacsacsőr jeleket érinti.

– Bizonyos mennyiségű (kb. 100-nál több) rövidítés esetén a program nem volt hajlandó lefordulni.
– A forrásfájl tartalmazott beégetett rövidítéseket, mint például a „CD”, amelyek kivételként kezelendők. (Mert gyakoribb esetben mondatvégek, és nem rövidítések.) Ez a lista nem volt bővíthető.
– Több különálló rövidítésfájl nem volt alkalmazható egyszerre.

Ezt a feladatot az új verzióban egy Python script végzi el, a fentiek figyelembevételével. Az *abbrev_en* szűrő egyedüli angol nyelvű szűrőként állt és csak egy tesztben és a rövidítéslistában különbözött az *abbrev* szűrőtől, ezért megszüntettem. Az angol nyelvű szövegek tokenizálásáról a következőkben lesz szó.

Egységesítésre kerültek a szűrőkben felvett definíciók, különös tekintettel a karakterosztályok neveire. Ennek célja, hogy nyelvfüggetlenebb legyen és többnyelvű[3] környezetben is képes legyen működni néhány definíció átírásával.

Az egységesítés lehetővé tette továbbá, hogy az eredeti XML-formátumtól eltérő, szabadon választott mezőelválasztó-jeleket lehessen használni, így mostantól ez a lehetőség is adott. A Unicode karaktertábla nagysága miatti szükséges változtatásként bevezettem, hogy csak egy meghatározott síkkészletet használjon a program, így a generált elemző kisebb és gyorsabb lesz. Ez a joker karakter („.” reguláris kifejezés), illetve a karakterlista-negáció („[ˆABC]” kifejezés) esetén fontos, mert ezeknél nagyon megnő a generált automata állapotszáma. Ennek elkerülésére a kiválasztott Unicode-síkok uniójával el vannak metszve ezek a kifejezések, és használjuk őket a későbbiekben. Ez a Quex beépített funkcióival valósult meg.

Az informatikai kifejezések között újak jelentek meg, mint például az IPv6 szabvány, vagy az ékezetes és Unicode-karaktereket tartalmazó, tetszőleges TLD-re végződő doménnevek. Ezek mind jobban bekerülnek a köztudatba, így az internetcímekkel kapcsolatos mintákat kibővítettem ennek megfelelően. Így már ezeket a tokenosztályokat is felismeri. Az egységesítés során a minták átnézése, javítása, egyszerűsítése is megtörtént. Ez főleg a tokenszűrőt érinti. A változások követéséhez, a Huntokenhez mellékelt Holt lelkek című Gogol-művet is felhasználtam, amit beépítettem állandó tesztnek a rövid specifikációtesztek mellé.

A tokenizálás nyelvfüggetlenségének érdekében két független verzió is készült az eredeti, csak minimális, a Quexre történő átültetéshez engedhetetlenül szükséges változtatásokat tartalmazó változat mellett, aminek célja az eredeti Huntoken funkcionalitások minél hűbb megtartása a Unicode karakterkódoláson.

A további két verzió egyike tartalmazza a fent említett változásokat, illetve a nyílt tokenosztályok ragozásának elemzésénél térnek el: az egyik változat megtartja az eredeti Huntokenben használt ragozásfelismerő eljárásokat és az MSD-kódolást. A másik változat egy beépülő morfológiai modulnak helyet ad, amely a beadott szó alapján meghatározza a lemmát és a címkéket. Ha más nyelven akarnánk használni a mondatra bontót, a megfelelő morfológia beépítésével erre is lenne lehetőségünk, mivel a legtöbb nyelvben már elérhető jó minőségű

[3] Itt elsősorban az európai nyelvekre gondolok. Például angol, német stb.: ezek speciális szóalkotó karaktereket tartalmazhatnak.

morfológiai elemző, de ezt az esetet gyakorlatban nem vizsgáltuk. Opcionálisan ez a lépés kihagyható, így a tokenekre bontás elemzés nélkül hajtódik végre, lehetőséget adva az utólag külön menetben történő elemzésre. Maguknak a tokeneknek a morfológiai elemzése elvégezhető lenne a tokenizálással egy menetben, de jogi okok miatt a Humor elemzőt [2] jelenleg még nincs mód beletenni a nyílt forráskódú rendszerbe.

A nyelvfüggetlenség mellett a különböző szakterületekre való könnyebb adaptálhatóság is a célok között volt. Például az orvosi szövegekben rengeteg rövidítés található, ellenben kevés az internetes cím.

A Quex képességeinek köszönhetően megoldható a különböző szűrők egy programba való lefordítása, illetve a szűrőnként az elemző bemenetének pufferből történő adagolása, ezzel további új távlatokat nyitva a program szélesebb körű felhasználhatósága előtt.

3. Eredmények

A cikkben bemutatott program, a PureToken egy olyan platform, nyelv- és címkekonvenció-független, Unicode-alapú mondatra bontó és tokenizáló eszköz, amely az elődjéhez képest számos kibővített funkciót tartalmaz, a kor elvárásainak megfelelően. Gyorsaságában és pontosságban ugyanazt a teljesítményt nyújtja, mint elődje, de néhány új tokenosztályt is felismer, és sokkal jobban testre szabható a működése. Az első tesztek is ezt igazolják. Egyetlen függősége a Quex- és a Python-környezet, valamint a C++ fordító. Célom, hogy széleskörű tesztelés után a visszajelzések alapján az esetleges új, vagy már a Huntokenben is meglevő hibákat javítsam, és szükség szerint karbantartást végezzek a programon, hogy minél több alkalmazási területen megállja a helyét.

Köszönetnyilvánítás

Köszönöm Németh Lászlónak, hogy megírta és szabadon hozzáférhetővé tette a Huntokent és hogy a fejlesztés során rendkívül hasznosnak bizonyult specifikációteszteket is írt hozzá.

Köszönjük a TÁMOP-4.2.1.B – 11/2/KMR-2011–0002 projekt részleges támogatását.

Hivatkozások

1. Miháczi A., Németh L., Rácz M.: Magyar szövegek természetes nyelvi feldolgozása. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). Szeged (2003) 38–43
2. Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Inquiries into Words, Constraints and Contexts. Stanford, California (2005) 150–157
3. http://quex.sourceforge.net/ Elérés 2012. 11. 30.

4. Csendes D., Hatvani Cs., Alexin Z., Csirik J., Gyimóthy T., Prószéky G., Váradi T.: Kézzel annotált magyar nyelvi korpusz: a Szeged Korpusz. Magyar szövegek természetes nyelvi feldolgozása. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). Szeged (2003) 238–247

Ismeretlen szavak helyes kezelése kötegelt helyesírás-ellenőrző programmal

Indig Balázs[1], Prószéky Gábor[1,2]

[1]Pázmány Péter Katolikus Egyetem, Információs Technológiai Kar,
MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter u. 50/a
indba@digitus.itk.ppke.hu, proszeky@itk.ppke.hu
[2]MorphoLogic, 1122 Budapest, Ráth György u. 36.
proszeky@morphologic.hu

Kivonat Napjainkban a közigazgatástól a könyvkiadásig jelentős szerepe van az összefüggő nagy terjedelmű szövegeknek. Ezek helyesírását meglehetősen nehéz és időigényes ellenőrizni, mert a szöveg vagy speciális tudást igényel egy adott szakterületről, vagy a nagy mennyiség miatt a javításra szánt idő válik jelentőssé. A gyakorlatban működő helyesírás-ellenőrző programok csak a szavak szintjére koncentrálnak, és legfeljebb egy-egy elgépelésre tudják figyelmeztetni a felhasználót, míg a létező, de a program számára ismeretlen, új szavakat, tulajdonneveket tévesen hibásnak jelölik. A cikkben bemutatásra kerülő módszer a nagyobb összefüggő szövegekben rejlő statisztikai sajátosságokra építve egy olyan helyesírás-ellenőrző módszert mutat be, amelynek célja a szövegösszefüggésekből kinyerhető ismeretlen, új, ám helyes szavak minél teljesebb felismerése, ragozási paradigmáik megsejtése, majd ezen szavak esetleges elgépeléseinek a javítása. A bemutatandó módszer lehetővé teszi hosszabb szövegek, például könyvek, intézményi dokumentumok egy lépésben történő gyors helyesírás-ellenőrzését.

1. Bevezetés

Az internet gyors átalakulásával és a számítógépek fejlődésével egyre szélesebb körben lehetővé válik, hogy mind nagyobb terjedelmű szövegeket hozzanak létre a felhasználók, és párhuzamosan elvárják, hogy a helyesírás-ellenőrző programok lépést tudjanak velük tartani. Ez nem kivitelezhető a hetvenes évek óta alig változó, szóról szóra haladó helyesírás-ellenőrző módszerekkel. Naponta új szavak, tulajdonnevek jelennek meg és keverednek a hagyományos szövegekkel, szófordulatokkal. Egyre több speciális területen rögzítik a szakszövegeket számítógépre, ahol egy általános helyesírás-ellenőrzőnek nincs lehetősége a szakterület speciális szavait ismerni, viszont az elgépelések esélye ugyanúgy fennáll.

Angol nyelven, ahol nincsenek túlsúlyban a ragozott szóalakok, a probléma kevésbé jelenik meg, viszont az erősen ragozó nyelvekben, mint a magyar, ez

határozottabban előkerül, ugyanis nemcsak az egyes új, helyesírás-ellenőrző környezetek által nem ismert szavakat „kellene" felismerni és javítani, hanem egyúttal ezek helyesen ragozott alakjait is. Bár az ismeretlen szavakról a gép jelenleg nem tudja eldönteni, hogy helyesek-e, egységesíteni tudja az írásmódjukat a statisztikailag releváns találatok alapján, illetve képes egy menetben csoportosítani és így egyszerre javítani vagy jóváhagyni több előforduló szóalakot a felhasználó kényelme érdekében. A módszer erősen támaszkodik arra, hogy egy szó jó alakja statisztikailag számottevően gyakoribb, mint az elgépelés. Természetesen ez a módszer a következetes helytelen írásmódot nem képes javítani.

Az alábbiakban ezen folyamat részleteit ismertetjük. Mi az általunk korábban kifejlesztett eszközöket használtuk, de a megoldás általánosabb, ezért a későbbiekben időnként tokenizálóként fogunk hivatkozni a PureTokenre [6], POS-taggerként fogunk hivatkozni a PurePOS-ra [3], és morfológiaként a Humorra [2].

2. A módszer

Az összefüggő szövegeknek sajátossága, hogy a bennük előforduló szavak a Zipf-törvény szerinti eloszlással rendelkeznek [5]. Megfelelő méretű összefüggő szövegeket választva a statisztika mind jobban előtérbe tolódik, a nyelvspecifikus ismeretek mellé. Ahogy az Kornai és társai cikkében [7] is szerepel, az internetről is legyűjthetők ilyen szövegek, amelyekből statisztikai jellemzők kinyerhetőek későbbi felhasználásra.

2.1. A statisztikai jellemzők kinyerése és felhasználása

Ezen jellemzők kinyeréséhez a rendelkezésre álló nyelvtechnológiai eszközök mindegyikét végigfuttatjuk a szövegen, és a mondatokra és tokenekre bontott szöveg szavaihoz szófaji címkéket és szótöveket rendelünk, majd egy hagyományos helyesírás-ellenőrzővel megjelöljük azokat a szavakat, amelyek ismeretlenek. Az így létrejött annotált szövegben – immár csak az ismeretlen szavakat tekintve – statisztikai sajátosságokat keresünk, amelyek segítségünkre lehetnek a szavak osztályozásában, illetve ajánlatgenerálásban. Ilyen jellemzők például:

- az egyes szóalakok gyakoriságai
- az ismeretlen szavak (POS által meghatározott) szótöveinek gyakoriságai
- a fentiek kombinációja.

A szótövek szerint csoportosított szóalakokból a magyar nyelv ragozási jellemzőinek és ezek összefüggéseinek ismeretében – amit a morfológia tartalmaz a beépített szótárban szereplő szavak esetén – kellő számú és minőségű különböző ragozott alak megléte esetén megállapítható egy ragozási paradigma, amire vizsgálhatóak a kevésbé gyakori szóalakok, így eldöntve, hogy ragozásuk egységes-e vagy sem, ezzel felismerve a helytelenül ragozott, esetleg elgépelt szóalakokat. Az így szerzett információval lehet felismerni és javítani a csak különféle elgépelt formában előforduló változatokat is, melyeket a hagyományos helyesírás-ellenőrzők a többi helytelen szóval egyetemben egységesen hibásnak jelölnek, további

elemzés nélkül. Egy másik probléma az ismeretlen, de elgépelt szavakhoz megfelelő ajánlások generálása, amit a fenti módon gyűjtött információkkal orvosoltunk.

Az ismeretlen szavak osztályát tovább bontva egy-egy szóalakot, illetve szótövet a gyakorisága alapján tekinthetünk „biztosan jónak" vagy pedig „ritkának"[1]. A „biztosan jó" szóalakokból és a gyakori szótövekből végezzük el a csoportosítást és a ragozási paradigma meghatározását. Ezek a szóalakok és a belőlük nyert információk segítenek a ritka szóalakokhoz ajánlások generálásában[2].

A hagyományos helyesírás-ellenőrzők így átalakíthatóak, hogy a megadott szavak és szótövek alapján paradigmát építve újraellenőrizzék az ismeretlennek jelölt szavakat, és szükség szerint ajánlásokat generáljanak hozzájuk a meglévő belső működés felhasználásával. Ezzel megbízható módon és teljesen automatikusan lehet bővíteni a helyesírás-ellenőrző és a morfológia szótárát. Emellett a felhasználó visszajelzést tud küldeni a fejlesztőknek, vagy egy központi adatbázisban gyűjtheti a kollaboratív munka eredményeit egy helyesírás-ellenőrző esetleges doménspecifikus tudásának felépítéséhez.

Az így kapott, osztályozott, javítási javaslatokkal ellátott szavak minden előfordulását a felhasználó könnyen, a teljes dokumentum átolvasása nélkül, mindössze a kritikus szövegkörnyezetre rápillantva, egy menetben kezelve képes javítani. A nyers szöveg mondatokra és tokenekre bontása közben ugyan elveszíti az eredeti formázást, de például dinamikus idővetemítéssel (DTW)[8] meghatározhatóak a szoros összefüggések (horgonyok) az eredeti szöveggel, arra az esetre, ha a javításokat nem szóalakonként egységesen, hanem a javítandó szavak környezetének függvényében kívánjuk elvégezni. Tipikusak az alábbi többértelműségek:

- *román*: a nemzetiség (román[MN][NOM]), a roma emberen (roma[FN][SUP])
- *rendben*: benne a rendben (rend[FN][INE]), rendben van (rendben[HA])
- *alma*: az állat alma (alom[FN][PSe3][NOM]), almafa (alma[FN][NOM])
- továbbá minden olyan toldaléksorra végződő alak, amelyek összetett szó utótagjaként is megjelenhet, például: *-ének: gyerekének, -ében: fejében, -ára: tanára, -inak: tanulóinak* [9]

2.2. A POS-tagger adaptálása a szöveghez a posteriori információkkal

A tokenizált szöveget a POS-taggernek átadva, az egyértelműen meghatározza a szavakhoz a lehetséges lemmákat a beépített morfológia segítségével.[3] Az ismert szavak esetén csak a néhány felkínált alternatíva közül kell választani a simított

[1] A gyakori, ugyanolyan módon történő elgépelést következetes hibának vesszük, és nem tudunk különbséget tenni következetes hibák szándékosságát illetően.

[2] Jelen mérésben csak egyszerű Damerau–Levenshtein távolságot [10] alkalmaztunk az ajánlások kereséséhez, de ez bővíthető több megszokott módszerrel is.

[3] Itt azt feltételeztük, hogy a helyesírás-ellenőrző nem szólista alapú, hanem morfológiát használ.

n-gram modell alapján. Ezzel szemben az ismeretlen szavak esetén a szótő és a szófaji címke meghatározása nem ilyen egyszerű: ekkor az ismeretlen szavakat egy ismeretlen szavakat elemezni képes modul, az ún. guesser megpróbálja megelemezni pusztán a beleépített nyelvi tudásra hagyatkozva. Az így kapott rengeteg elemzés közül kell kiválasztania a megfelelőt az egyértelműsítőnek, amely csak a lokális, n-gram modellt, illetve a mondatszintű beam search megoldást veszi figyelembe [3]. Más szóval: nem használja ki a nagy terjedelmű összefüggő szövegekben rejlő globális információkat. A POS-tagger hatékonyságának javítására olyan módszert dolgoztunk ki, amely a feldolgozott szöveg a posteriori információi alapján támogatja a feldolgozást: a szöveg feldolgozása közben a guesser által az egyes szavakhoz generált lehetséges lemmák közül a szóhoz tartozó címkének megfelelőkből mindig a globálisan leggyakoribbat választjuk. Ezzel előállítunk egy, a lemmák gyakorisága szerint rendezett listát, amelyből a megfelelően választott előfordulási küszöb fölötti, így gyakori szótöveket beadhatjuk a programnak listaként, hogy válassza ki azokat a lemma–címke párokat, amelyeknél a szótő szerepel a listán, ha van ilyen. Ezzel redukálja a lehetőségek számát, majd az így leszűkített halmazból kiválasztja a végleges verziót. Az eljárástól azt várjuk, hogy az egy szótőre visszavezetett ismeretlen szavak száma nő, ezzel pedig a helyes szótövek száma az ismeretlen szóalakok egészét tekintve arányosan javul.

3. Eredmények

A módszer hatékonyságát egy elméletileg csak helyes szavakat tartalmazó regényen (Orwell: 1984) és az internetről legyűjtött újságcikkekből és cikksorozatokból álló hasonló méretű korpuszon vizsgáltuk, a Szeged 2.0 korpuszt [4] használva nyelvi modellként. Az ellenőrzés során egy egyszerű heurisztikával szűrést végeztünk. Az eredetileg kapott adatokat az 1. táblázatban sz.e., a szűrés utániakat sz.u. jelzi. A szűréssel a statisztikából kivettük az egyértelműen önálló toldalékként azonosítható szavakat (pl. „-nak”) és az olyan szavakat, amelyek nem tartalmaztak legalább négy egymás melletti betűt (pl. „TU-154”, „MiG-24”). Ezáltal azt reméljük, hogy az „igazi” szavak és elgépeléseik jobban előtérbe kerülnek.

1. táblázat. A korpuszok adatai.

	1984		*Újságcikkek*	
	sz.e.	*sz.u.*	*sz.e.*	*sz.u.*
Tokenek:	99913	50586	74053	40716
Tokenek (egyedi):	20393	18211	20916	18465
Szegedben nem szereplő:	1149	1058	10001	8965
Szegedben nem szereplő (egyedi):	956	881	8321	7582
Humorban nem szereplő:	301	283	1431	1224
Humorban nem szereplő (egyedi):	181	168	1029	886
Humorban és Szegedben sem szereplő:	217	199	1362	1166
Humorban és Szegedben sem szereplő (egyedi):	129	116	992	859

2. táblázat. Példa a szavak gyakoriságára.

szó	**gyakoriság**	**szótő**
Obama	40	Obama
Obamaáról	1	Obamaá
Obamáék	1	Obamá
Obama-kormány	1	Obama-kormány
Obamának	3	Obam
Obamának	3	Obamá
Obamára	1	Obamá
Obamáról	3	Obam
Obamáról	3	Obamá
Obamát	5	Obam
Obamát	5	Obamát
Obamával	1	Obamával

A 2. táblázatban látható, hogy a globális információ nélküli program nem tudta megtalálni a kapcsolatot a különböző szóalakok között. Az elgépelés belesimul a helyes, ismeretlen alakokba. A szöveg méretétől függően érdemes beállítani a gyakorisági küszöböt, amitől egy szótő, illetve szóalak helyesnek számít. Mi a mérés során az alábbi paramétereket választottuk: szógyakoriság ≥ 2, tőgyakoriság ≥ 3 és Damerau–Levenshtein távolság $= 1$.

3. táblázat. Eredmények.

	1984	*Újságcikkek*
Szótőváltozás:	34	65
Szótőváltozás (egyedi):	19	48
Gyakori lemmák száma:	14	55
Gyakori szóalakok száma:	40	51
Paradigmák száma:	17	58
Ajánlások száma:	3	8

4. táblázat. Jó paradigmák.

1984	
szótő	
beszélír	
jó szóalakok	**ritka szóalakok**
beszélírba	beszélírja
beszélírral	beszélírtól
beszélír	
beszélírt	

Újságcikkek	
szótő	
Obama	
jó szóalakok	**ritka szóalakok**
Obamának	Obamáék
Obamáról	Obamára
Obamát	Obamával
Obama	

A ragozási paradigmák, amelyek a 4. táblázatban is láthatóak, akkor tekinthetőek jónak, ha megfelelő számú és minőségű olyan szóalakot találunk, amelyek alkalmasak az egyértelmű osztályozásra, így a bizonytalan, ritkább alakok ellenőrzésére. Rossz egy paradigma, ha a szótő sok ritka szóalak csoportosításaként, illetve ha túl kevés szóalak gyakori előfordulása miatt lett gyakori. Ez utóbbiak is természetes módon előfordulnak a szövegben. Az ajánlások a jónak tekintett szavak alapján történtek (5. táblázat).

5. táblázat. Ajánlások

Újságcikkek	
hibás szóalak	**ajánlás**
BruxInfo	Bruxinfo
Gingrics	Gingrich
Mtelekom	MTelekom
Obamaáról	Obamáról
Osama	Obama
Sandber	Sandberg
stent	sztent
Unicredit	UniCredit

1984	
hibás szóalak	**ajánlás**
aszondom	Aszondom
beszélírja	beszélírba
jógondoló	jógondol

Látszik, hogy érdemes egy már meglévő helyesírás-ellenőrző program motorját használni, mert különben a primitív algoritmusnak köszönhetően olyan hamis ajánlások is születhetnek, amelyek elkerülhetők lennének.

A vizsgált korpuszokon a hagyományos helyesírás-ellenőrző programok által helytelenül hibásnak jelzett szavak aránya csökkent, és néhány esetben sikerült a hibásan gépelt ismeretlen szavakat helyesre javítani, minimális zajarány mellett.

4. További kutatási irányok

A módszer jelen pillanatban önmagában még nem alkalmas automatikus helyesírás-ellenőrzésre, de a kutatásnak ez a kezdeti fázisa azt mutatja, hogy az új módszer használatával a teljes ellenőrzési folyamat a szöveg méretének növelésével egyszerűbbé és gyorsabbá válik.

Az újfajta helyesírási hibák ember által felügyelt javításával pedig már most is kielégítő eredményt kapunk, a felhasználó pedig az összefüggő szövegek javítását gyorsabban, kényelmesebben és pontosabban tudja végezni. További kutatásainkban a módszer alábbi alkalmazási lehetőségeit vizsgáljuk:

- a helyesírás-ellenőrző tudásának bővítése hatékonyan;
- egy erre a célra hasznos elgépelésszótár automatikus építése;
- felhasználók közötti kollaboráció a helyesírás-ellenőrzésben és javításban megosztott lexikonnal;
- mindezek által gyors doménadaptáció elérése.

A felsorolt folyamatok jelenleg meglehetősen emberigényesek, de a javasolt módszer az egységnyi idő alatt feldolgozható szöveg mennyiségét egyértelműen növeli.

Köszönetnyilvánítás

Köszönjük a TÁMOP-4.2.1.B – 11/2/KMR-2011–0002 projekt részleges támogatását.

Hivatkozások

1. Miháczi A., Németh L., Rácz M.: Magyar szövegek természetes nyelvi feldolgozása. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). SZTE, Szeged (2003) 38–43
2. Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Inquiries into Words, Constraints and Contexts. Stanford, California (2005) 150–157
3. Novák A., Orosz Gy., Indig B.: Javában taggelünk. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011). SZTE, Szeged (2011) 336–340
4. Csendes D., Hatvani Cs., Alexin Z., Csirik J., Gyimóthy T., Prószéky G., Váradi T.: Kézzel annotált magyar nyelvi korpusz: a Szeged Korpusz. Magyar szövegek természetes nyelvi feldolgozása. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003). SZTE, Szeged (2003) 238–247
5. Zipf, G.: Selective Studies and the Principle of Relative Frequency in Language. Cambridge, Mass (1932)
6. Indig B.: PureToken: egy új tokenizáló eszköz. In: Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013). SZTE, Szeged (2013) 305–309

7. Kornai, A., Halácsy, P., Nagy, V., Oravecz, Cs., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In: Proceedings of the 2nd International Workshop on Web as Corpus (WAC '06). Association for Computational Linguistics, Stroudsburg, PA, USA (2006) 1–8
8. Bellman, R., Kalaba, R.: On adaptive control processes. IRE Transactions on Automatic Control, Vol. 4, No. 2 (1959) 1–9
9. Novák A., M. Pintér T.: Milyen a még jobb Humor. In: IV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2006). SZTE, Szeged (2006) 60–69
10. Damerau, F. J.: A technique for computer detection and correction of spelling errors. Commun. ACM, Vol. 7, No. 3 (1964) 171–176

A ReALIS statikus interpretációjának kísérleti implementációja

Károly Márton[1]

Pécsi Tudományegyetem, Rektori Hivatal, „Science, Please!" Projektiroda
ReALIS Elméleti és Számítógépes Nyelvészeti Kutatócsoport
7624 Pécs, Vasvári Pál u. 4.
harczymarczy@gmail.com

Kivonat: Kutatócsoportunk célja a ReALIS statikus interpretációjának bemutatása. Ez a ReALIS definíciójában szereplő külvilági entitások és magrelációk Prolog-tényekké való direkt leképezésével, valamint az elmék működésének (szabályokkal történő) szimulálásával történik. A rendszerben a külvilágba ágyazottan több interpretálói entitás (elme) is szerepel, melyek szintén Prolog-tényekké leképezve tartalmazzák a külvilágnak, ill. más interpretálók elméinek (szükségképpen torz) projekcióit. A torzulás az információ téves vagy nem teljes, sőt ellentmondó voltából eredhet. E keretek között a program feladata jelenlegi formájában egy kiterjesztett klasszikus igazságértékelésen túl az, hogy egyes mondatokról eldöntse, hogy azok az interpretálók szájából elhangozhatnak-e. A program a feldolgozás során kiírja, hogy mely információkhoz nyúlt e kérdések megválaszolása során. A rendszer jelenleg csak egy nagyon egyszerű generatív szintaxist használ, az alsóbb nyelvi szintekkel való összeépítés és a teljes rendszer integrációja későbbi, komoly feladat. Ugyanakkor a több elme szimultán kezelése már a projekt hosszú távú céljainak szem előtt tartásával történik. A program SICStus Prolog 4.2.1-ben készült.

Kulcsszavak: statikus interpretáció, igazságértékelés, elmereprezentáció

1. Bevezetés: az elmélet működésének bemutatása

A ReALIS statikus interpretációja voltaképpen egy általánosított igazságértékelés, melynek feladata a klasszikus logikából ismert igazságértékelésen túl az is, hogy az állítások vagy megnyilatkozások preszuppozícióit – előfeltevéseit – is vizsgálja. Ily módon kezelhetővé válnak olyan, a klasszikus logikában értelmezhetetlen mondatok

[1] A szerzőt e cikk alapjait jelentő kutatásaiban a TÁMOP-4.2.1.B-10/2/KONV/2010/ KONV-2010-0002 (A Dél-dunántúli régió egyetemi versenyképességének fejlesztése) támogatta, beleértve ebbe a konferencián való részvételt is.

is, mint amilyen *A jelenlegi nepáli király kopasz*.[2] Sőt innen már csak egy lépés, hogy állítások helyett bármely más megnyilatkozás pragmatikailag helyes vagy hibás voltát is vizsgáljuk, vagyis azt, hogy megtörténhet-e, helyénvaló-e az adott kontextusban, vagy sem.

2. A program leírása

2.1. Input

A program inputja jelenleg egy „egyszerű”, azaz néhány igéből, névszóból, valamint modális szerkezetből (*úgy-tudja-hogy* stb.) álló mondat. Ennek – egyszerű generatív szintaxissal történő – elemzése során a program „lefordítja” az input tényállításban szereplő mondatot a ℜeALIS nyelvezetére, majd az (előre betáplált) adatbázis alapján meghatározza, mely interpretáló szájából hangozhat el az adott mondat, valamint egy általánosabb klasszikus igazságértékelést is ad. Ez utóbbinak része természetesen az is, hogy az intenzionális, bizonyos elmeállapotot (vagy ha majd lesz dinamikus interpretációnk, annak megváltozását) kifejező igék vizsgálata esetén az érintett elme tartalmát, annak eventualitásait megvizsgáljuk. A horgonyzás azonban ebben az esetben is a beszélő elméje alapján történik, így *a jelenlegi nepáli király* esetén azt a választ kell adja a rendszer: probléma van a (ki)horgonyzással. (Egészen pontosan: a kihorgonyzás szükséges voltából következik maga a *Nepálnak jelenleg van királya* preszuppozíció. Ez természetesen nem igaz, így a külvilágból nézve *A jelenlegi nepáli király kopasz* állítás nem értékelhető ki a klasszikus kétértékű logika eszközeivel. A horgonyzás problémája csak egy olyan általánosabb, kiterjesztett igazságértékelés segítségével vizsgálható, mint amilyen a ℜeALIS statikus interpretációja.

2.2. Az adatbázis és a program működése

A programban explicit módon definiálva vannak az entitások (`entity`, azonosítóval és 1-es típusszámmal jelölve) és a magrelációk (`corerel`, azonosítóval és 2-es típusszámmal, e példában a „valaki a főnöke valakinek” jogi kategóriát írjuk le magrelációval), a programban ez utóbbiak elemeit tekintjük infonoknak [1:142]. Az adatbázis alatt még mindig Prolog-tényeket kell érteni, a rendszer végső kiépítettségében azonban SQL-adattáblákból történik majd a szükséges tényadatok beolvasása:

```
entity(3,1,'Péter'). entity(4,1,'Józsi').
entity(5,1,'Juli'). entity(6,1,'Géza').
corerel(8,2,'főnöke').
corerelval(8,2,-1,1,[[3,4],[5,6]].
```

[2] Nepál 2008. május 28. óta köztársaság, előtte királyság volt. Ennek tudása azonban az átlagos interpretáló számára nem triviális, ezért a valóságban is előfordulhat, hogy Nepál utolsó uralkodójáról mint jelenlegi királyról beszéljen valaki.

A magrelációk értéktáblájában (`corerelval`) már provizorikusan felvettem egy (a jelenhez relatív) időintervallum-paramétert is, ez akkor jut majd szerephez, ha korábbi időpillanatokat vizsgálunk vagy egy múltbéli történésnek a jelenre való kihatását. Ennek lehetőségét az implementáció következő lépésében teremtjük majd meg.

Ezen kívül a [2]-ben elméleti szinten leírt adatstruktúrák (a cikkben `alpha`, `lambda` expliciten szerepel) időközben célszerűen módosított változataira alapozunk: a gyakorlati megvalósítás során az `alpha` nem választható el a morfoszintaxistól, mert a horgonyzásokat a nyelvi pillérek legitimálják oly módon, hogy az ige, az esetkeretek és (a későbbi megvalósítás során) a szórend meghatározzák a hatásláncot és az operátor-hatóköri sorrendet. Ezek alapján pedig már horgonyozhatunk. (A biztosnak tekintett tudást egyszerűsítésképpen az adott elme gyökérvilágába tartozónak tekintjük, melynek referensei ki vannak horgonyozva a külvilágba.) Most *A főnököm felesége csinos* mondat példáján bemutatjuk a program működését (a *felesége* magreláció felépítésének kitalálását pedig az olvasóra bízzuk):

```
Péter: A főnököm felesége csinos.
e: főnöke(Péter, Józsi) gyökérvilág OK
I: főnöke(Péter, Józsi) külvilág OK
e: úgy_gondolja(Péter, e: felesége(Józsi, Helga)) OK
e: csinos(Helga): interpretálói elmék vizsgálata...
e: úgy_gondolja(Péter,e: csinos(Helga))
e: úgy_gondolja(Juli,e: csinos(Helga))
e: úgy_gondolja(Mari,e: csinos(Helga))
e: úgy_gondolja(Józsi,e: csinos(Helga)) OK
e: felesége(Józsi, Helga) gyökérvilág OK
I: felesége(Józsi, Helga) külvilág OK
A mondat elhangozhat. Az állítás igaz.

Józsi: A főnököm felesége csinos.
Hiba: e: főnöke eventualitás Józsi elméjében nincs.
A preszuppozíció nem teljesül („lódítás”?).

Juli: A főnököm felesége csinos.
e: főnöke(Juli, Géza) gyökérvilág OK
I: főnöke(Juli, Géza) külvilág OK
e: úgy_gondolja(Juli,e: felesége(Géza,Mari))
I: felesége(Géza, Bori) külvilág
Hiba: probléma a „felesége” horgonyzásával! A
preszuppozíció téves.
úgy_gondolja(Péter,csinos(Bori)) Kevés!
Az állítás hamis.
úgy_gondolja(Juli,csinos(Mari))
úgy_gondolja(Géza,csinos(Mari))
úgy_gondolja(Péter,csinos(Mari))
úgy_gondolja(Józsi,csinos(Mari)) OK
A mondat elhangozhat.
```

Itt *A főnököm felesége csinos* mondat három különböző elemzését mutatjuk be. Öt releváns interpretálóból álló beszélőközösséget feltételezünk: Péter, Józsi, Mari, Juli, Géza (Helga és Bori nem tagjai a közösségnek). A *főnöke* és a *felesége* a jog (munkaszerződés, ill. házasság) eszközeivel egyértelműen körülírható, tehát értelmezésünk szerint azok a külvilágban létező **magrelációk**, melyek példányai az **infonok** (még időparaméter nélkül).

Az első eset az, amikor Péter főnöke létezik, ő Józsi, neki van egy felesége, Helga. Péter tudja, hogy Helga valóban Józsi felesége, és csinosnak gondolja. A mondat ezért a grice-i maximák szerint elhangozhat, sőt a külvilág szemszögéből is igaz. Péter elméjében eventualitásként, de ami még fontosabb, a külvilágban infonként is szerepelnek a *főnöke* és *felesége* relációk, és ellenőrizhető, hogy Péter és Józsi, valamint Józsi és Helga is a megfelelő relációban vannak. (A `corerelval` első négy argumentuma: azonosító, típus és két időparaméter: a *felesége* reláció eleme valamikor a múltban létrejött, és a jövőben feltehetően is még létezni fog). (Az egyelemű reláció természetesen itt is a halmaznak felel meg.)

A preszuppozíció ellenőrzése után az igazságértékelés következik. A magreláció igaz voltát elegendő volna a külvilágban ellenőrizni, a *csinos* azonban nem az, hanem egy ún. **kvantált intenzionális** reláció, amely csak a beszélőközösség elméjében létezik. (Egyszerű – tehát nem kvantált – intenzionális ige pl. a *bevesz vki vmit* [1:70–73]). A kvantált intenzionális relációknál a beszélőközösség összes tagjának elméjét meg kell vizsgálni, hogy szerepel-e bennük egy megfelelő világocskában a *csinos(Helga)*. (A beszélőközösséghez tartozást amúgy egy pontosabb modellben szintén magrelációval írhatjuk le.) A *csinos* predikátumot ezek után akkor fogadjuk el igaznak, ha a többség úgy gondolja, megkerülve így a *mennyire csinos?* kérdés feltevéséből adódó, a rendszer fuzzyfikálását megkívánó problémakört. Helga esetében az öt beszélőből négy csinosnak gondolja Helgát, tehát az állítás *igaz*.

Ugyanezt Józsira alkalmazva látjuk, hogy a *főnöke* reláció rá vonatkozóan nem tartalmaz adatokat: Józsi vállalkozó, saját cége van, egyik alkalmazottja történetesen Péter. Ezért az ő főnökére vonatkozó mondat grice-i értelemben nem hangozhat el (ha nem tudják a partnerek, hogy Józsinak nincs főnöke, akkor **blöff**ről beszélünk). *A jelenlegi nepáli király kopasz* mondattól ez annyiban különbözik, hogy Józsi biztosan tudatában van annak, hogy amire referál, az nem létezik. A külvilág felől nézve mindazonáltal ez a mondat **rosszul formált** és **nincs igazságértéke.**

Juli a főnökét, Gézát gyakran látja Marival, ezért őt a feleségének gondolja. Mivel még nem régóta dolgozik Géza cégénél, nem tudja, hogy Mari valójában csak a főnöke új szeretője, a feleségét valójában Borinak hívják. A horgonyzással ezért probléma van, de **ettől még a mondat grice-i értelemben elhangozhat** Juli szájából mindaddig, amíg meg nem tudja az igazságot (ti. hogy Géza már egy ideje csalja Marival Borit). Juli ugyanis a mondatot igaznak hiszi, sőt mivel Mari a beszélők többségének álláspontja szerint is csinos, azt mindenki el is hiszi, aki nincs tisztában azzal, hogy Géza felesége valójában Bori.

A külvilágban ez a mondat természetesen **hamis**, mert Borit csak Péter gondolja csinosnak, a beszélőközösség többi tagja nem. De az a helyzet is előállhatna, amennyiben Bori is csinos volna, hogy a mondat tulajdonképpen **igaz**, csak Juli és a többiek nem ugyanarra a személyre gondolnak. A *Géza felesége Mari* preszuppozíción alapuló horgonyzás problematikus voltát természetesen jelzi a program.

A horgonyzást, így a preszuppozíciót tehát **mindig a beszélőnél keressük**, és csak ez után vetjük össze a külvilággal. Ha mindent rendben találunk, a mondat jól formált. Vizsgálnunk kell, hogy 1. létezik-e az, amiről állítottunk valamit, 2. ugyanaz van-e a külvilágban, mint a beszélő elméjében. Ha az 1. feltétel nem teljesül (pl. azért nem, mert Nepál már nem királyság), akkor a mondat már ki sem értékelhető, bár ettől még lehetséges, hogy elhangozhat grice-i értelemben. Ha igen, de a 2. feltétellel gond van, vagyis az előfeltevés a beszélő elméjében téves, akkor a mondat ugyan elhangozhat, de a program jelzi, hogy annak vélelmezett igazságértéke téves feltevésen alapul. Ez után történik meg a külvilág szemszögéből történő igazságértékelés. Ez utóbbi úgy történik, hogy kívülről indul el az elemzés befelé az elmék λ-szintjein keresztül. Így a *Mari azt-gondolja-hogy Péter nős* mondat is kiértékelhető (intenzionalitás kezelése).

3. Összegezés. A közeljövőben megoldandó problémák

A program jelen állás szerint néhány extenzionális (*felesége, főnöke*) és intenzionális (*csinos, azt-gondolja-hogy*) predikátumot képes kezelni. Ezen kívül még további entitásokat és néhány relációt is felvettem, köztük olyanokat is, mint pl. a *férfi, nő, nős* egyváltozós reláció. Ez utóbbinak lényege abban áll, hogy közvetett úton akarjuk visszavezetni a *felesége* extenzionális relációra (egzisztenciális kvantálás). Ez közös háttértudás révén lehetséges, melynek helye szintén az interpretálói elmében van, ezért nincs szükség arra, hogy a külvilágban *ha-akkor* típusú szabályokat vezessünk be. Sőt van olyan vélekedés is, hogy az entitásokat kivéve minden az emberi elmén belül játszódik le, tehát magrelációkra sincs szükség. Ez persze azt jelenti, hogy az összes elmét végig kellene vizsgálni az igazságértékeléshez minden esetben, ráadásul az információk **hitelességének** problémaköre sem kerülhető meg. További gond az ún. „elfekvő iratok" esete: a külvilág tartalmazhat olyan relációt, amely aktuálisan egyetlen elmében sincs benne. De persze, ha a programban nem is, a ℜeALIS eredeti definíciójában szerepel a PERCEIVE magreláció, ennek révén az „elévült", elfelejtett információ bármikor kinyerhető a külvilágból – de csak onnan.

A másik megoldandó probléma az **időbeliség** kezelése: bár az adatszerkezetet valamelyest megalapoztuk, az „elévülés" nem csak az elmék felejtése miatt következhet be, hanem azért is, mert alapos okunk van azt hinni, hogy a *nepáli király* esetéhez részben hasonlóan az információ már érvényét vesztette akkor is, ha a külvilágban valójában nem változott semmi (különbség: Nepál időközben köztársaság lett). Az információ relevanciája ugyanis az idővel csökken, és előbb-utóbb alá megy a grice-i maxima érvényesüléséhez szükséges határnak – de ez mindig csak az adott predikátum ismeretében mérhető. Egy halotti toron például csak akkor mondhatjuk, hogy *Péter* (az elhunyt) *nős volt*, ha az elhalálozás időpontjában volt felesége, ha előtte évekig elváltan élt, akkor nem. A *Péter hazament* esetén pedig az „elévülés" ideje nem más, mint a cselekvés eredményszakaszának vége. Ha azt mondjuk, hogy *Péter hazament*, akkor vélhetően azért tettük, mert Péter valószínűleg éppen otthon van. Ez azonban további cikkek tárgyát képezhetné, ahogy az ezt kezelni tudó program bemutatása is.

Hivatkozások

1. Alberti G.: ℜeALIS. Akadémiai Kiadó, Budapest (2011)
2. Alberti, G., Károly, M.: The Implemented Human Interpreter as a Database. In: Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Párizs (2011) 379–385
3. Alberti, G., Károly, M., Kleiber, J.: From Sentences to Scope Relations and Backward. In: Sharp, B., Zock, M. (eds.): Natural Language Processing and Cognitive Science. Proceedings of NLPCS 2010. SciTePress, Funchal, Madeira, Portugália (2010) 100–111
4. Alberti G., Kilián I.: Vonzatkeretlisták helyett polaritásos hatásláncсaládok – avagy a ℜeALIS σ függvénye. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia, MSZNY 2010. SzTE Informatikai Tanszékcsoport (2010) 113–126
5. Károly M.: Interpretáció és modalitás – avagy a ℜeALIS λ-függvényének implementációja felé. In: MSZNY 2011. SzTE Informatikai Tanszékcsoport (2011) 284–296

A szövegkorpuszok szókincsének összehasonlítása szótári címszójegyzék felhasználásával – neologizmusok és archaizmusok detektálása

Kiss Gábor[1], Kiss Márton[1]

[1] TINTA Könyvkiadó, Budapest
{kissgabo, kissmarci}@tintakiado.hu

Kivonat: A Magyar Történeti Korpusz (TK) és a Magyar Nemzeti Szövegtár (MNSz) összehasonlítása egy más irányú lexikográfiai feladat elvégzésének melléktermékeként jött létre a TINTA Könyvkiadóban. Az elsődleges feladat az Értelmező szótár+ (ÉrtSz+ [1]) címszavainak gyakorisági mutatóval való ellátása volt. A gyakorisági mutatók meghatározásához felhasználtuk mindkét magyar szövegkorpuszt. Az elsődleges feladat elvégzése során megvizsgáltuk, hogy az ÉrtSz+ 15.850 címszava előfordul-e, és ha igen, milyen gyakran a fenti két magyar szövegkorpuszban külön-külön. A két korpuszból kinyert gyakorisági adatok segítségével (súlyozást is alkalmazva) állapítottuk meg az egyes címszók gyakorisági osztályát, azaz soroltuk be az 5 gyakorisági osztály valamelyikébe.

1 A vizsgálat előzménye

A Magyar Történeti Korpusz (TK) és a Magyar Nemzeti Szövegtár (MNSz) összehasonlítása egy más irányú lexikográfiai feladat elvégzésének melléktermékeként jött létre a TINTA Könyvkiadóban. (Mint ismeretes, a TK-t a Magyar Nagyszótár munkálatai során építették fel, és ez a korpusz 18., 19. és 20. századi magyar szövegrészleteket tartalmaz, míg az MNSz-t a 20. század végén keletkezett elsősorban sajtónyelvi, irodalmi szövegek alkotják.)

Az elsődleges feladat az Értelmező szótár+ (ÉrtSz+ [1]) címszavainak gyakorisági mutatóval való ellátása volt. A gyakorisági mutatók meghatározásához felhasználtuk mindkét magyar szövegkorpuszt[1]. Az utóbbi évtizedekben a nemzetközi szótárirodalomban az angol értelmező szótárak nyomán elterjedt a címszavak gyakoriságának jelölése. A magyar szótárirodalomban a Magyar értelmező kéziszótár (ÉKsz.[2] [2]) közli elsőként a címszavak gyakoriságát.

Az ÉrtSz+ sajátosan izgalmas színt képvisel a magyar, de a nemzetközi szótárkínálatban is, mert erőteljesen túllép az értelmező szótár szokásos funkcióin. Az ÉrtSz+ a szómagyarázó funkció mellett – mint a szótár az alcímében is jelzi – Értelmezések, példamondatok, szinonimák, ellentétek, szólások, közmondások, etimológiák, nyelv-

[1] Ezúton is köszönetet mondunk az MTA Nyelvtudományi Intézetének a két korpusz szokásos felhasználási módját meghaladó vizsgálat engedélyezéséhez.

használati tanácsok és fogalomköri csoportok szerint is átfogó módon dolgozza fel címszóállományát. Sőt a szótár még a címszó gyakoriságát is feltünteti egy ötfokozatú skálán.

Mint említettük, az ÉrtSz+ címszavai gyakoriságának meghatározásához felhasználtuk a TK-t és az MNSz-t. A feladat elvégzése során megvizsgáltuk, hogy az ÉrtSz+ 15.850 címszava előfordul-e, és ha igen, milyen gyakran a fenti két magyar szövegkorpuszban külön-külön. A két korpuszból kinyert gyakorisági adatokat – leegyszerűsítve mondva – átlagoltuk, majd súlyozást is alkalmazva állapítottuk meg az egyes címszók gyakorisági osztályát, azaz soroltuk be az 5 gyakorisági osztály valamelyikébe. A kapott eredmény elemzését követően, a gyakorisági mutatót néhány esetben szubjektív nyelvérzékünk alapján módosítottuk.

2 A vizsgálat

Szövegkorpuszok összehasonlítására nincs általánosan elfogadott módszer. A matematikai logika nyelvére lefordítva: két nagy, de véges számú, ismétlődő diszkrét elemeket is tartalmazó halmazt kell összevetni. A halmazokban nemcsak az egyes elemek, azaz szavak megléte vagy hiánya a fontos, hanem az is jellemző, hogy egy-egy szó hányszor fordul elő egy-egy szövegkorpuszban.

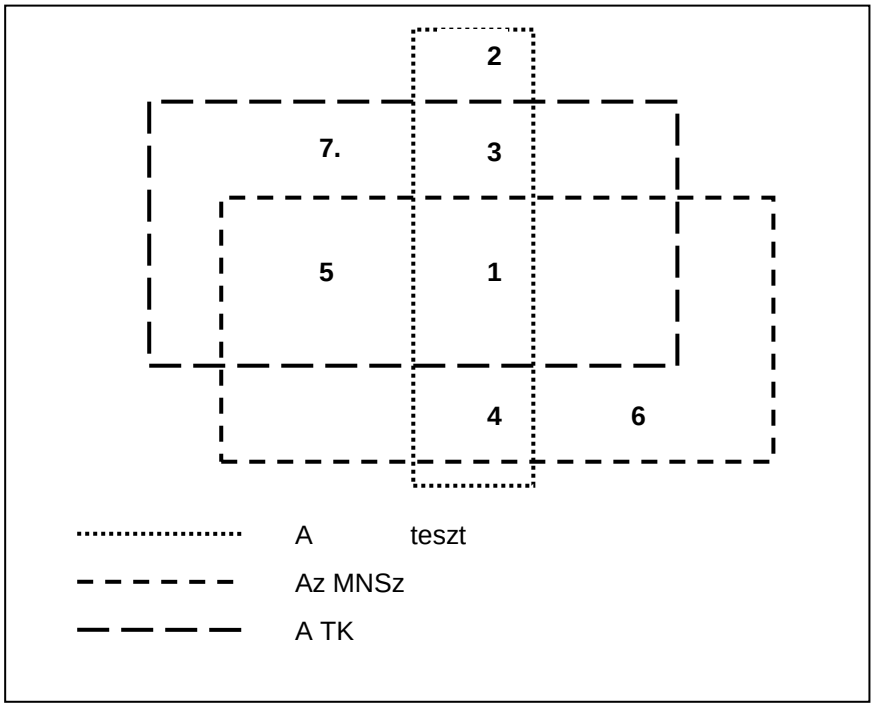
Mivel a gyakorlatban szinte lehetetlen két vizsgált nagy szövegkorpusz minden egyes szavának az összevetéséből keletkezett eredmény kiértékelése, az ÉrtSz+ címszavainak gyakorisági mutatójának a meghatározása megmutatott egy praktikusnak és helyesnek tűnő eljárást ahhoz, hogy két vagy több szövegkorpusz szókincsét miként lehet összevetni. Ugyanis a munkálat rávilágított arra, hogy egy jól megválasztott minta, vagy más néven tesztszólistának a szavait felhasználhatjuk a korpuszok szókincsének összevetésére, összehasonlítására. Természetesen ennek a tesztszólistának a hossza az összehasonlítandó korpuszok szókincséhez képest nem lehet se túl kicsi, se túl nagy. Az ÉrtSz+ címszavainak gyakorisági osztályokba sorolása során úgy tapasztaltuk, hogy a szótár 15.850 címszavának a két korpuszban való előfordulása és az előfordulások gyakorisága jól jellemzi és sajátosan leírja a fenti két korpuszt. Így az ÉrtSz+ címszójegyzékét joggal tekinthetjük vizsgálatunk tesztszólistájának.

Az ÉrtSz+ címszavait különös gonddal válogatták össze a szerkesztők. A címszólista összeállításáról így vallanak: „Milyen szavakat tartalmaz címszóként a szótár? Elsősorban az úgynevezett alapszókincs elemeit, amelyeket a leggyakrabban és legtöbben használunk, ezek között egyaránt vannak fogalomszók (...) és formaszók (...). Az alapszavak mellett sok a tantárgyi szakszó, olyanok, amelyekkel a diákok az irodalom-, a nyelvtan-, a történelem-, a matematika-, a fizikaórán találkozhatnak. (...) Megtalálható a szótárban több olyan régi szó, amelyek már csak irodalmi és történelmi szövegekben olvashatók (...). ezeken kívül vannak benne új, legtöbbször a modern technikával kapcsolatos szavak (...) Helyet kaptak a szótárban olyan szavak is, amelyeket Magyarországon nem használunk, de a határon túli magyarok életéhez hozzátartoznak.'' [1: VIII].

Két korpusz (MNSz és TK) szókincsének (szavainak) és a tesztszólistaként használt ÉrtSz+ címszavainak összevetése során a következő elméleti esetek lehetségesek:

1. A tesztszólista egy adott szava mindkét korpuszban előfordul valahányszor.
2. A tesztszólista egy adott szava nem fordul elő egyik korpuszban sem.
3. A tesztszólista egy adott szava a TK korpuszban előfordul valahányszor, míg az MNSz-ben nem.
4. A tesztszólista egy adott szava az MNSZ-ben előfordul valahányszor, míg a TK-ban nem.
5. A tesztszólistában nem szereplő szó mindkét korpuszban előfordul.
6. A tesztszólistában nem szereplő szó az MNSZ-ben előfordul, de a TK-ban nem.
7. A tesztszólistában nem szereplő szó a TK-ban előfordul, de az MNSZ-ben nem.

A Magyar Nemzeti Szövegtár (MNSz), a Magyar Történeti Korpusz (TK) és a tesztszólista szavainak lehetséges viszonya:



Vizsgálatunk körébe természetszerűleg csak a 1−4. pontok tartoztak, az 5−7. pontok esetei kívül estek figyelmünkön. Mérésünket 2006 májusában végeztük. Ebben az időben az MNSz 111.746.000 szövegszó nagyságú, míg a TK 8.897.000 szövegszó terjedelmű volt. A mérés elején az ÉrtSz+ címszavai közül eltávolítottuk az 1-nél nagyobb homonimaindexszel ellátott címszavakat. A homonimák nélküli tesztszólista az eredetileg 15.810 címszóból álló lista helyett 15.010 szó hosszúságú lett. Ezt követően egy kis robotprogram lekérdezte a tesztszólistát alkotó 15.010 szó előfordulási gyakoriságát a két korpuszban. A MNSz és a TK korpuszok a szövegszavaik korábbi

morfológiai elemzése során − ha kis mértékben is − eltérő morfológiai elemzési technikát alkalmaztak, ezért az egyes esetekhez tartozó szavak számát 10-es értékre kerekítettük.

3 A vizsgálat eredményei

3.1 Hasonlósági mutatók

Számszerűsítve a következő eredményt kaptuk a teszt szólista 15.010 szava és a vizsgált két korpusz szavainak viszonyára:

1. A tesztszólista 15.010 szavából 14.290 szó előfordult mind a TK-ban, mind az MNSz-ben; 95,20%
2. A tesztszólista 15.010 szavából 45 szó nem fordult elő egyik szövegkorpuszban sem; 0,30%
3. A tesztszólista 15.010 szavából 670 szó csak az MNSz-ben fordult elő, a TK-ban nem szerepelt; 4,46%
4. A tesztszólista 15.010 szavából 5 szó csak a TK-ban fordult elő, az MNSz-ben nem szerepelt; 0,03%.

Az adatokból látszik, hogy a két magyar szövegkorpusz szókincse hasonló, jelentős átfedést mutat egymással. A tesztszólistának használt ÉrtSz+ címszójegyzéke a szótár jellegéből és funkciójából adódóan jobban illeszkedik a MNSz-hez, mint a TK-hoz.

3.2 Neologizmusok

Az MNSz jellegéből adódóan joggal feltételezhetjük, hogy azok a szavak, amelyek csak az MNSz-ben fordulnak elő, jól jellemzik a 20. század végét, illetve az ezredfordulót, és így ezek a magyar szókincs neologizmusai közé tartoznak. Feltételezésünk igazolására közreadjuk annak az 50 leggyakoribb szónak a listáját, amelyek az MNSz-ben előfordulnak, de a TK-ban nem találhatók meg (a szó melletti szám a szó MNSz-beli előfordulását mutatja).

Neologizmusok, az MNSz leggyakoribb korfestő szavai:

közszolgálati 6764
munkáltató 6263
honlap 5383
euró 4515
közterület 4415
sportág 4204
CD 3890
piacgazdaság 3743
e-mail 3728
szoftver 3611
környezetvédelem 3592
digitális 3590
szia 3475
drog 3399
parkoló 3373
atomerőmű 3217
elsőfokú 3184
internet 3057
közalkalmazott 3008
foci 2541
bevásárlóközpont 2476
kosárlabda 2323
világháló 2294
társasház 2234
tömegközlekedés 2051
informatika 1993
globalizáció 1833
videó 1726

bróker 1566
multi 1378
természetvédelem 1136
AIDS 1079
telefax 1060
hobbi 1055
tizenéves 970
rajzfilm 955
mobiltelefon 912
tévécsatorna 873
interaktív 860
kemping 852
versenyszféra 803
éllovas 794
elektronika 789
sci-fi 761
papírforma 755
hardver 735
lakópark 693
akciófilm 666
sikerdíj 636
bulvárlap 635

3.3 Archaizmusok

Ezt követően felmerül a kérdés, hogy a fenti neologizmusokhoz hasonlóan a magyar nyelv archaizmusait is kigyűjthetjük-e a két korpusz szóanyagának az összevetéséből? Mivel vizsgálatunkban a TK-ban nem találtunk jelentős számban olyan szavakat, amelyek csak abban fordulnak elő, és nem találhatók meg a MNSz-ben – direkt módon nem gyűjthetünk archaizmusokat a TK-ból. Azonban adódik a gondolat, hogy talán archaizmusnak tekinthetők azok a szavak is, amelyek arányaiban jóval többször fordulnak elő a TK-ban, mint az MNSz-ben. A gondolat életrevalónak tűnik, bizonyításképpen alább közreadjuk az első 50 olyan szót, amelyek jelentősen többször fordulnak elő a TK-ban, mint az MNSz-ben (a szót követő szám az előfordulási arányt mutatja).

Archaizmusok, azok a szavak, amelyek arányaiban a TK-ban jelentősen többször fordulnak elő, mint az MNSz-ben:

aprószentek 15
asztag 12
beszély 15
billikom 45
borong 13
bőszült 20
burnót 24
csepű 14
csibuk 28
csigáz 99
csöngettyű 13
dicső 13
divatozik 22
dragonyos 16
epeszt 36
fejkötő 20
findzsa 21
gondola 16
hajdankor 23
hanga 23
hazámfia 64
hevenyében 15
honfi 28
honn 47
horgany 50
ispán 16
játszi 26
kebel 16
kegyed 25
komika 30
komorna 24
kopja 20
korhely 13
ködmön 13
mál 28
málé 11
mente 14
messzely 28
midőn 34
nadály 18
okuláré 11
orozva 27
pacsuli 20
patvar 14
sarjadék 17
sólya 14
süveg 13
szövétnek 11
szüle 18
tekintetes 18
téns 54
tragika 21
urambátyám 20
vágta 23
várta 109
vitézkötés 13

3.4 A magyar szókincs magja

Mind a két vizsgált korpuszban előfordul a tesztszólista szavainak 95,2%-a. Mint láttuk, nem mindegy azonban, hogy a közös szavak előfordulásának mi az aránya. A következőkben közreadunk mintaképpen 50 olyan gyakori szót, amelyek előfordulási aránya megegyezik vagy közel azonos mindkét szövegkorpuszban. Az 50 mintaszó mindegyikét az ÉrtSz+ leggyakoribb címszavai közül választottuk, azaz mindegyik az első gyakorisági kategóriába tartozik a szótár öt kategóriájából. A szavak mögött álló szám az MNSz-ben és a TK-ban lévő előfordulások aránya. Ha a szám nagyobb egynél, akkor arányaiban az MNSz-ben fordult elő többször a szó, ellenkező esetben a TK-ban gyakoribb.

Megállapíthatjuk, hogy ezek a szavak fontosak, a magyar szókincs középpontjában, magjában helyezkednek el, hiszen használatuk több mint két évszázadon át gyakori és állandó intenzitású a korpuszok adatai alapján.

Állandó intenzitású és gyakori szavak a magyar szókincs magjából:

ad 1,13
csinál 0,99
elmegy 0,87
esik 1,15
este 0,92
férfi 1,17
fiatal 0,90
gondol 1,03
győz 0,86
három 0,99
hat (szn) 1,02
hisz 0,85
hoz 1,00
idő 1,13
ismer 0,89
jut 1,06
katona 0,88
két 1,17
kevés 0,91
könnyű 0,83
könyv 1,03
magas 0,92
magyaráz 1,00
mond 0,83
nagyon 1,18
nap 1,03
négy 1,17
nehéz 0,93
név 1,20
nyár 1,09
ok 1,13
olvas 0,88
óra 1,17
orvos 1,02
rendel 0,85
rossz 1,03
sok 1,03
század 0,89
széles 0,89
szeret 1,10
szó 0,92
szolgál 0,92
tart 1,18
tartozik 1,04
teremt 1,15
tud 1,19
út 1,12
város 1,13
vesz 0,95
víz 0,87

4 Összegzés

Összegzésként elmondhatjuk, hogy újszerű vizsgálatunk bebizonyította, hogy két szövegkorpusz szókincsének összehasonlítása eredményesen elvégezhető egy kisebb alkalmas tesztszólista – akár egy szótár címszójegyzékének − segítségével.

Két különböző jellegű szövegkorpuszból hatékonyan gyűjthetők ki egy megfelelő tesztszólista segítségével a korpuszokra külön-külön is jellemző szavak, a mi esetünkben neologizmusok és archaizmusok. A kigyűjtött neologizmusok a 20. század végé-

nek, az ezredfordulónak a korfestői, míg az archaizmusok jól jellemzik a 18. század vége, illetve a 19. század magyar szókincsét.

Ugyanakkor megállapítható, hogy a két korpusz közös elemei közül azok, amelyek előfordulása magas és az előfordulások aránya közel azonos az egyes korpuszokban, a magyar szókincs magját képezik, így köznyelvi szótárak címszójegyzékének összeállításához eredményesen használhatók.

A két magyar szövegkorpusz összehasonlítása során nyert eredmények rávilágítanak arra is, hogy több és különböző típusú magyar szövegkorpuszra lenne szükség, mert a szövegkorpuszok összevetésével jellegzetes csoportokat alkotó szavak gyűjthetők egybe többé-kevésbé automatikusan. A korpuszokból kigyűjtött szócsoportok alkalmas kiindulásai lehetnek a magyar szókészlet különböző szempontú szótári munkálatainak.

Hivatkozások

1. Előry V. (főszerk.): Értelmező szótár+. Értelmezések, példamondatok, szinonimák, ellentétek, szólások, közmondások, etimológiák, nyelvhasználati tanácsok és fogalomköri csoportok. TINTA Könyvkiadó, Budapest (2007)
2. Pusztai F. (főszerk.): Magyar értelmező kéziszótár[2]. Akadémiai Kiadó, Budapest (2003)

Morfológiai egyértelműsítés nyelvfüggetlen annotáló módszerek kombinálásával

Laki László János, Orosz György

MTA-PPKE Magyar Nyelvtechnológiai kutatócsoport,
Pázmány Péter Katolikus Egyetem, Információs Technológiai Kar
1083, Budapest, Práter utca 50/a
e-mail:{laki.laszlo, oroszgy}@itk.ppke.hu

Kivonat Írásunkban megvizsgálunk két szófaji egyértelműsítő modult, s arra következtetésre jutunk, hogy bizonyos esetekben a két rendszer hibái nagyon távoliak. Bemutatjuk, hogy egy esetleges kombináció milyen eredményekkel kecsegtethet, illetve ismertetünk két egyszerű összetételi technikát, melyek segítségével készített nyelvfüggetlen rendszer a morfológiai tudást használó társával pontosság tekintetében versenyképes.

1. Bevezetés

A szófaji egyértelműsítés a számítógépes nyelvfeldolgozás egyik alapfeladata. A feladat megoldására számos szabadon elérhető nyelvfüggetlen rendszer használható, melyek többsége valamilyen statisztikai tanuló algoritmust használ. Egy-egy eszköz nagyon alacsony hibarátája több, mint kívánatos, hiszen egy szövegfeldolgozási láncban a többi elemző algoritmus ennek kimenetére épít, ezt használja.

Jelen írásunkban először ismertetünk két szófaji egyértelműsítő rendszert: a PurePos [1] eszközt és egy statisztikai gépi fordításon alapuló PoS-taggert [2]. Közelebbről megvizsgálva az általuk hibásan osztályzott szavakat, azt találtuk, hogy a rendszerek által vétett hibák közötti átfedés nagyon alacsony. Ebből az észrevételből kiindulva, megvizsgáltuk, hogy milyen lehetőségek nyílnak a két rendszer tudásának kombinálására. Megmutatjuk, hogy csupán a két nyelvfüggetlen rendszer kombinációját használva, jobb eredményt érhetünk el, mint egy harmadikkal való egyszerű szavazásos kombinációt használva. Eredményeinkből az is kiolvasható, hogy a prezentált nyelvfüggetlen metódus címkézési pontosságban versenyképes lehet a PurePos morfológiai elemzővel segített változatával.

2. A használt eszközök

Magyar nyelvre egy szabadon elérhető statisztikai alapon működő, de mégis hibrid rendszer a PurePos, mely integrált morfológiai elemzőt tartalmazó rejtett Markov-modellezésen alapuló teljes egyértelműsítő rendszer. A rendszer a Brants

[3] és Halácsy et al. [4] által ismertetett algoritmusokra épít, különös tekintetettel a morfológiai elemző teljes integrációjára. Az egyszerű simított trigram modellnek köszönhetően magas precizitással és alacsony tanítási idővel rendelkezik. Az eszköz Java nyelven íródott, így szükség esetén könnyen módosítható. Megmutattuk [1], hogy azon esetekben, amikor lehetőség van morfológia használatára, kis méretű tanítóanyag esetén is jelentős növekedést ér el mind a szófaji címkézés, mind pedig a lemma egyértelmű meghatározása esetén is.

Egy korábbi írásunkban [2] megvizsgáltuk a statisztikai gépi fordítórendszer (SMT) szófaji egyértelműsítő és szótövesítő eljárásként való alkalmazhatóságát (HuLaPos). Itt sikerült megmutatnunk, hogy minimális előfeldolgozással viszonylag kis méretű tanítóhalmaz esetén is jó minőségű PoS-tagger állítható elő. Ez többnyire annak volt köszönhető, hogy a szófaji egyértelműsítés feladata nagyságrendekkel kisebb komplexitású a szóösszekötő rendszer számára, mint egy természetes nyelvi fordítási feladat, valamint a kifejezés alapú gépi fordítórendszer döntése során képes figyelembe venni a szavak mindkét oldali környezetét is. Az SMT-módszer leggyengébb pontja a szótárban nem szereplő szavak elemzése. Egy szógyakoriságon alapuló módszerrel sikerült az OOV[1] szavak környezetének súlyát megnövelni és ezzel a rendszerre gyakorolt negatív hatását csökkenteni.

Cikkünkben felhasználjuk még az OpenNLP [5] eszközkészletben elérhető maximum entrópiás és perceptron tanulásos algoritmusokat is. Az említett eljárások nagy népszerűségnek örvendenek, mivel a tanuló algoritmusában használt jellemzők könnyen adaptálhatóak egy-egy új feladatra. Ezen módszerekre igaz még, hogy a nagy számosságú jellemzőhalmaz miatt a tanítási idejük nagyságrendekkel nagyobb rejtett Markov-modellezésen alapuló társaiknál.

A PurePos esetén láttuk a morfológiai tudás nagyon értékes tud lenni – különös tekintettel agglutinativ nyelvek esetén – de sajnos csak korlátozott számú nyelvre érhető el szabadon morfológiai elemző. Továbbá egy új elemző létrehozása nagyon időigényes, és nyelvész szakértők bevonását igényli, így felmerülhet az igény olyan általános célú módszerekre, melyek csupán a tanító halmazt használva magas pontossággal képesek szófaji egyértelműsítésre.

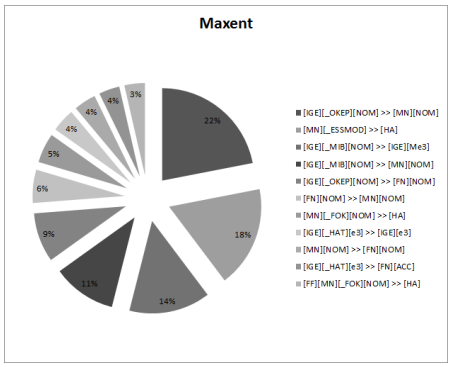
3. Az összetett rendszer

3.1. Motiváció

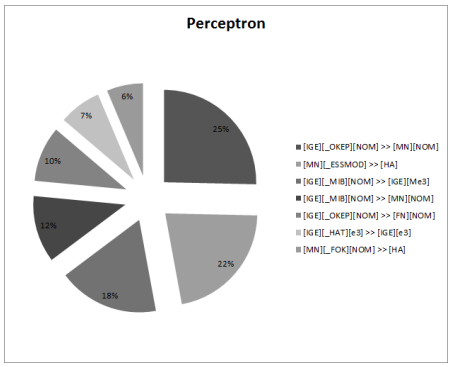
Megvizsgáltuk a négy rendszer hibáit[2] (1. és 2. ábra), s azt találtuk, hogy bár a PurePos pontossága általában magasabb társainál, de az általa vétett hibák átfedése az SMT-alapú HuLaPos rendszerrel alacsony átfedésben van. Ezen kívül nagy számban előfordulnak olyan hibák is, melyeket az OpenNLP valamely eljárása javított helyesen. Viszont az is megfigyelhető, hogy a maxent és perceptron tanulásos algoritmusok hibái jelentős részben egybeesnek. Továbbá, az Orosz

[1] Az egyértelműsítő által korábban nem látott események.

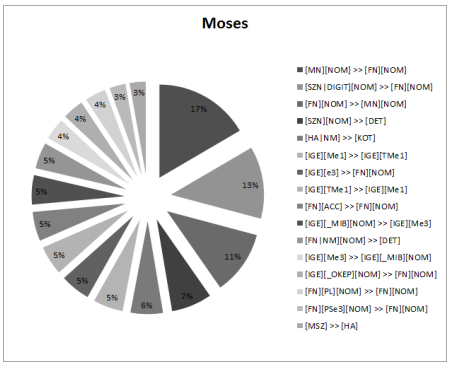
[2] A prezentált hibák az egyes taggerek által vétett hibatípusok legjellemzőbb 40%-át tartalmazzák `helyes címke >> tippelt címke` formátumban.



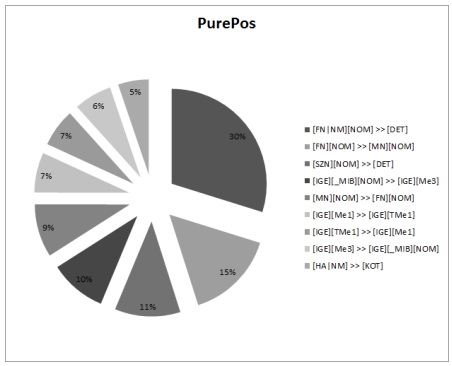
(a) A maxent tanulás gyakori hibái



(b) A perceptron tanulás gyakori hibái

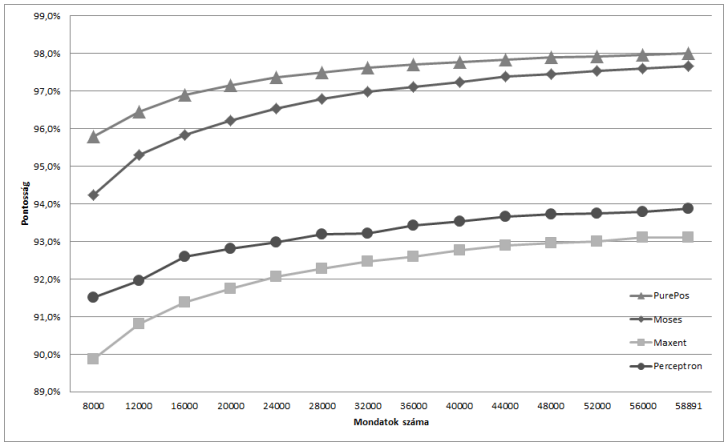


(c) A HuLaPos rendszer gyakori hibái



(d) A PurePos rendszer gyakori hibái

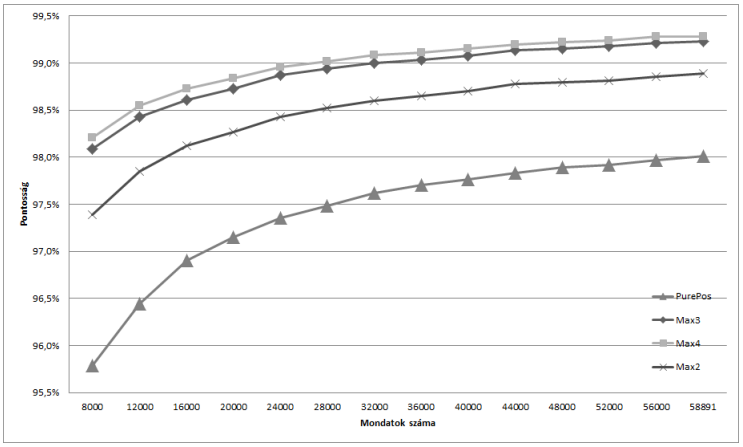
1. ábra: A szófaji egyértelműsítő rendszerek leggyakoribb hibáinak összetétele.



2. ábra: A szófaji egyértelműsítő rendszerek eredményessége a tanítóanyag méretének függvényében.

által ismertetett eszköz hibáinak legnagyobb része a határozott névelő – mutató névmás; számnév – határozatlan névelő ambiguitási osztályok rossz címkézése, míg a gépifordító-rendszer sokszor a számára ismeretlen szavakat nem tudja a megfelelő morfoszintaktikai osztályba sorolni.

A fenti hibaanalízisből merítve megvizsgáltuk, milyen maximális együttes tudással rendelkezhet egy olyan rendszer, mely az egyes rendszerek összetételéből állhat. Vizsgálatunkat a Szeged Korpuszon [6] HuMor [7,8] tagekre konvertált változatán végeztük, annak 10%-át elkülönítve tesztelési célra, míg a többin inkrementálisan tanulva vizsgáltuk, hogy hogyan változik az egyes címkézők pontossága a tanítóanyag méretének változásával. A 3. ábrán megfigyelhető, hogy a két, a három, illetve a négy rendszer szignifikánsan jobban teljesít a többinél és hogy legalább egyike a fennálló hibák legalább 44,24; 61,26; 63,90 százalékáról rendelkezik helyes információval.



3. ábra: A PoS taggerek aggregált szófaji egyértelműsítő képessége.

A továbbiakban a jelen előzetes felmérésben legjobban teljesítő kettő, illetve három rendszer összetételével foglalkozunk.

4. Kombináció

Két – hagyományos értelemben vett – osztályozó algoritmus kombinációja esetén a kutatónak „csupán” az egyes esetekhez tartozó megfelelő jellemzőhalmazt és az összetételi algoritmust kell megválasztania. Esetünkben – bár a PoS taggelés is tekinthető osztályozási problémának – a helyzet összetettebb, mert az egyes események – ami a szó és a hozzá tartozó morfoszintaktikai címke – nem függetlenek egymástól. Továbbá ezen elven alapul a legtöbb szófaji egyértelműsítő

módszer is, nevezetesen egy mondathoz tartozó legvalószínűbb címkesorozat keresése az egyes szavakhoz tartozóak helyett. Így két út áll előttünk: az összetétel alapjaként tekinthetjük az egyes mondatokat, így a két rendszer valódi kimenete között választva, vagy a tokenszintű címkézési hibákat javítjuk. A hibaanalízisben részletezetteknek megfelelően, jelen írásunkban a második eshetőséget vizsgáljuk.

Az elsőként elkészített kombinációs technika egy egyszerű többségi szavazáson alapuló algoritmus volt. Páratlan számú résztvevőt használva a három előzetesen legjobban teljesítőt választottuk: a PurePos, az SMT-alapú és a perceptron tanulásos algoritmust használtuk. A szavazás azon fázisában, amikor a három rendszer nem tud dönteni, a legjobbnak vélt PurePos rendszer szavazatát tekintjük helyesnek. Ezzel az egyszerű módszerrel relatív 12,05%-os javulást értünk el szószintű pontosságot tekintve.

Következő lépésként az előzetesen két legjobban teljesítő rendszert kombináltuk. Két osztályozó között a többségi szavazás nem tud működni, így az alábbi algoritmust alkalmaztuk: a két címkéző mondatonként végzi az annotálást, majd a szavakat egyesével megvizsgálva, ha egy szónál egyetértés van a taggerek között, akkor azt elfogadjuk, ellenben egy gépi tanulásos algoritmus a korábban látott hibák alapján eldönti, hogy mely egyértelműsítő szavazatát részesítse előnyben. A hagyományos szófaji címkék tanulásához szükséges tanítóhalmaz mellett, elkülönítettünk egy ezzel diszjunkt, a címkézők hibáinak tanulásához szükségeset is. Így kutatásunkat a Szeged Korpusz egy részén képzett 50000 mondat méretű tanítóanyagon végeztük, melyen felül még 5000 mondatot használtunk a másodszintű tanításra, melyhez az alábbi szószintű jellemzőket találtunk: a szó, a megelőző szó, a kettővel megelőző szó, következő szó, kettővel rákövetkező szó, PurePos-címketipp, HuLaPos-címketipp, PurePos címketippje a következő szóra és a megelőző szóra, tartalmaz-e kötőjelet, tartalmaz-e pontot, nagybetűvel kezdődik-e, maximum 10 hosszú suffixek.

1. táblázat: A egyes kombinációs algoritmusokkal elért pontosság.

	NaiveBayes	PRISM	IB1
Pontosság	98,48%	98,23%	**98,51%**

2. táblázat: A kombinációs módszerek eredményessége a kiindulási rendszerek tükrében.

	HuLaPos	PurePos	PurePos(M)	Max2	Comb3	IB1
Pontosság	97,40%	**97,82%**	98,53%	98,77%	98,08%	**98,51%**

A 8000 mondatból álló optimalizálásra szánt halmazon – ami 133752 tokenből és 3566 másodszintű eseményből[3] áll – megvizsgáltunk, miként teljesít néhány,

[3] Azon esetek, amikor a két tagger tippje nem egyezik.

a WEKA [9] keretrendszeren keresztül elérhető algoritmus, melyek eredményességéről a 1. táblázatban számolunk be. A legjobbnak vélt IB1 [10] algoritmussal való kombináció pontosságát egy új teszthalmazon összevetettük a már meglévő egyértelműsítőink eredményével (2. táblázat). (A táblázatban a PurePos(M) a morfológiai tudást alkalmazó rendszert, a Max2 a HuLaPos és a nyelvfüggetlen PurePos maximális tudását, a Comb3 a három rendszerből álló egyszerű szavazást, míg az IB1 a két elemből álló összetételt jelöli.) Azt találtuk, hogy az így készített – előzetes nyelvi tudást nélkülöző – rendszer, szószintű pontosságot tekintve megelőzi a három rendszerből álló egyszerű többségi szavazást, sőt hibák számát tekintve versenyképes a PurePos morfológiát tartalmazó változatával is. A teszthalmazon mérve csupán relatív 1,22%-os a két eljárás közti hibák relatív különbsége.

5. Összefoglalás

Cikkünkben bemutattunk két csak statisztikai módszeren alapuló szófaji egyértelműsítő rendszert és azok jellemző hibáit, melyekből kiindulva megvizsgáltuk azok kombinációjának lehetőségét. Megmutattuk, hogy a két eszköz együttes tudása jelentősen meghaladhatja az önálló rendszerekét. Az elérhető tudás kihasználása érdekében tett erőfeszítésünk eredményeként ismertettünk két összetételi technikát. Az utóbb prezentált rendszer nemcsak hogy meghaladja a három rendszerből álló szavazásos összetétel eredményeit, de bemutattuk, hogy egyes esetekben olyan más eljárásokkal is versenyképes, melyek integrált nyelvi tudással dolgoznak.

Eredményeink bizakodásra adnak okot, így jövőbeni tervünk, hogy megvizsgáljuk, miként lehetséges a két algoritmusra alkalmazott összetételi technikát kiterjeszteni három vagy több rendszerre.

Hivatkozások

1. Orosz, Gy., Novák, A.: PurePos – an open source morphological disambiguator. In Sharp, B., Zock, M., eds.: Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science, Wroclaw (2012) 53–63
2. Laki, L.J.: Investigating the Possibilities of Using SMT for Text Annotation. In: SLATE 2012 - Symposium on Languages, Applications and Technologies, Braga, Portugal, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012) 267–283
3. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the sixth conference on Applied natural language processing. Number i, Universität des Saarlandes, Computational Linguistics, Association for Computational Linguistics (2000) 224–231
4. Halácsy, P., Kornai, A., Oravecz, Cs.: HunPos: an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Prague, Czech Republic, Association for Computational Linguistics (2007) 209–212
5. Baldridge, J., Morton, T., Bierner, G.: The OpenNLP maximum entropy package (2002)

6. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In: Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora LINC 2004 at The 20th International Conference on Computational Linguistics COLING 2004. (2004) 19–23
7. Novák, A.: Milyen a jó humor? In: Magyar Számítógépes Nyelvészeti Konferencia 2003, Szeged (2003) 138–145.
8. Prószéky, G., Novák, A.: Computational Morphologies for Small Uralic Languages. In: Inquiries into Words, Constraints and Contexts., Stanford, California (2005) 150–157
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. ACM SIGKDD Explorations Newsletter **11**(1) (2009) 10
10. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6**(1) (1991) 37–66
11. Kuba, A., Felföldi, L., Kocsor, A.: POS tagger combinations on Hungarian text. In: 2nd International Joint Conference on Natural Language Processing, Jeju Island, Republic of Korea, Association for Computational Linguistics (2005)
12. Vincze, V., Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: Proceedings of the Seventh conference on International Language Resources and Evaluation. (2010)

Anonimizálási gyakorlat? – Egy magyar korpusz anonimizálásának tanulságai

Mátyus Kinga

MTA Nyelvtudományi Intézet
1068 Budapest, Benczúr u. 33.
matyus.kinga@nytud.mta.hu

Kivonat: A Budapesti Szociolingvisztikai Interjú (BUSZI) második változata, vagyis a BUSZI-2 mintegy 100 órányi felvétele számos kutatásra ad lehetőséget. Az interjúk irányított beszélgetései sok személyes adatot tartalmaznak, melyek segítségével az adatközlők azonosíthatók. Azonban ahhoz, hogy a kutatók a korpuszhoz hozzáférjenek, az abban lévő érzékeny adatokat kezelni kell. A BUSZI-2 ötven interjúját az adatvédelmi törvény (1992. évi LXIII.) alapján, a korpusz jellegzetességeit is szem előtt tartva, illetve a nemzetközi gyakorlatra építve anonimizáltuk.

1 Bevezetés

Az MTA Nyelvtudományi Intézet Élőnyelvi Munkacsoportja 1985-ben kezdte meg egy olyan nagyszabású, beszélt nyelvet vizsgáló, abban az időben nemzetközi szinten is kimagasló kutatás megtervezését, melynek célja az volt, hogy az eddigi írott korpuszokon alapuló leírásokat jelentős nagyságú beszélt nyelvi korpusz elemzéséből nyert adatokkal kiegészíthessék, módosíthassák (részletesen lásd [5]).

A Budapesti Szociolingvisztikai Interjú (BUSZI) elnevezésű projektnek 2 korpusza készült el. A BUSZI-2 felvételeit 1987-ben rögzítették 50 adatközlővel, kvótaminta alapján: 5 különböző foglalkozási csoport 10-10 adatközlőjével készítettek 1,5–2,5 órás felvételeket. A BUSZI-3,-4 200 adatközlővel készült, rétegzett mintavétel alapján [8]. A korpuszok OTKA és AKP pályázati támogatásokkal[1] jöttek létre, a projektum vezetője Kontra Miklós volt. A BUSZI-2 ma már regisztrált kutatók számára hozzáférhető,[2] a BUSZI-3, -4 feldolgozása folyamatban van.

Mind a BUSZI-2, mind a BUSZI-3, -4 két nagy részből állt: kártyás feladatokból/ tesztfeladatokból és irányított beszélgetésekből.[3] Az irányított beszélgetések a Labov [6] által kidolgozott módszert követve az adatközlőknek nem megkomponált interjúnak, hanem inkább oldott beszélgetésnek tűntek, ennek is köszönhető, hogy rendkívül sok személyes adatot tartalmaznak. Ahhoz, hogy ehhez a korpuszhoz bármilyen kutató

[1] Részletesen lásd: http://buszi.nytud.hu/a-buszi-rol
[2] http://buszi.nytud.hu/kutatni-szeretnem-a-buszi-t
[3] Részletesen lásd: http://buszi.nytud.hu/a-buszi-rol/az-interjuk-felepitese

hozzáférjen, a benne található személyes adatokat kezelni kell. Jelen tanulmány a BUSZI-2 korpusz anonimizálását mutatja be.

1.1 Jogi szabályozás Magyarországon

A személyes adatok védelméről és a közérdekű adatok nyilvánosságáról Magyarországon az 1992. évi LXIII. törvény rendelkezik. Ennek értelmében a BUSZI-2 mind személyes (név, azonosító jel, fizikai, fiziológiai jellemzők stb.), mind pedig különleges adatokat (faji eredet, kisebbség, vallás, egészségi állapot stb.) tartalmaz. A kutathatósághoz szükséges, hogy az adatok és az érintett közti kapcsolat helyreállíthatóságát megszüntessük.

2 Anonimizálás más korpuszokban

Azok a beszélt nyelvi korpuszok, melyek azzal a céllal készültek, hogy szélesebb közönség számára elérhetőek legyenek, a következő gyakorlatokat követik: hozzájárulási nyilatkozatot kérnek az adatközlőktől, hogy a korpuszt közzé lehessen tenni. Az írott szövegek anonimitása könnyen biztosítható azzal, hogy törlik, vagy megváltoztatják például a neveket a korpuszban, azonban a hangfájlok anonimizálása már számos vélemény szerint nem lehetséges [4]. A Routledge Handbook of Corpus Linguistics szerzői hasonló alapelveket fogalmaznak meg: a tradicionális megközelítés szerint az adatközlők anonimitását hangsúlyosan szem előtt kell tartani. Ehhez a beszélők nevét és más részleteket meg kell változtatni, vagy teljesen törölni kell. Az anonimizálás bizonyos gyakran használt szavakra, kifejezésekre, sőt témákra is kiterjedhet, amelyek bármilyen módon felismerhetővé tehetik az adatközlőt. Az anonimitás még problematikusabb a hang- és videofelvételek esetén. Az egyéni hang, mint egy ujjlenyomat, azonosítja a beszélőt. A hang eltorzítása azonban akadályozhatja a korpusz fonetikai/fonológiai kutathatóságát – ezért nem ajánlott [1]. A személyes adatok kezelését szabályozó törvények minden országban különbözőek.

Lou Bernard a British National Corpus (BNC) anonimizálása kapcsán a következő lehetőségeket vázolta: 1) a nevet egyszerűen törölni kell, vagy XXXX-szel helyettesíteni, 2) egy kódot kell alkalmazni, amely minden Maggie betűsort egy kódra (XYZ12-re) cserél, vagy 3) egy szótárban a hasonló neveket hasonlóan fordítják – pl. a Maggie-nek Susan, a Jonesnak Brown lehetne a megfelelője [3].

A következőkben röviden bemutatjuk néhány korpusz gyakorlatát.

2.1 British National Corpus (BNC)

Azzal a céllal készült a korpusz, hogy a gyűjtött anyagot széles körben elérhetővé tegyék. Teljes anonimitást és bizalmas adatkezelést ígértek az adatközlőknek. (A beszélt részben, amely a korpusz 10%-át jelenti, 1) az adatközlők mikrofont viseltek, és felvették saját beszélgetéseiket, illetve 2) terepmunkások különböző műfajú beszédeket rögzítettek.)

A neveket és címeket törölték a korpuszból és a kapcsolódó dokumentumokból, helyette a <gap> címke áll magyarázattal, pl. <gap desc="name" reason= "anonymization"/>. Azt az ötletet, hogy a neveket kóddal vagy egy nyelvileg hasonló névvel helyettesítsék, praktikussági szempontok miatt elvetették.

2.2 Newcastle Electronic Corpus of Tyneside English (NECTE)

Amikor a BUSZI projektum az 1980-as évek végén elkezdődött, a ma is hatályos adatvédelmi törvény még nem élt. Hasonló volt a helyzet a Newcastle Corpus of Tyneside English (NECTE) esetében is. Beal [2] beszámol arról, milyen etikai és jogi feladatokat kellett megoldaniuk egy olyan korpusz közzététele során, amely még az 1998-as brit adatvédelmi törvény előttről származik. A NECTE egy hagyatékkorpusz, amely két részből áll: Tyneside Linguistic Survey (TLS) (1969-ből), és a Phonological Variation and Change Project (PVC) (1994-ből). A projekt célja az volt, hogy minél szélesebb körnek elérhetővé tegyék e két korpuszt. Mivel a TLS korpusz adatközlőinek anonimitást ígértek, a kutatók minden nevet töröltek a felvételekről és az átiratokról [2].

A TLS interjúk „érzékeny" témákat is érintenek (egészség, vallás, politika, szakszervezet), nem lenne elfogadható az a megoldás, hogy a korpuszt az interneten teszik hozzáférhetővé. Ezért azoknak a kutatóknak, akik a NECTÉ-vel szeretnének dolgozni, ki kell tölteniük egy nyomtatványt, s aláírva vissza kell küldeniük a központba.

2.3 BEA – magyar spontánbeszéd-adatbázis

A magyar Beszélt Nyelvi Adatbázis (BEA) esetében, az Amerikai Egyesült Államokbeli gyakorlathoz hasonlóan az adatközlőknek alá kell írniuk egy hivatalos hozzájárulási nyilatkozatot, és a kutatás során a személyes adatokat az interjútól külön kezelik, a kutatók a személyes adatokhoz nem juthatnak hozzá – kivéve természetesen a kutatáshoz szükséges adatokat, mint például a magasság és a kor.

3 A BUSZI-2 anonimizálása

A BUSZI-2 esetében az összes interjút kódolták, illetve lejegyezték, és az átiratokat kétszer ellenőrizték, hangfelvételek digitalizálták. Oravecz Csaba és Sass Bálint a szöveges lejegyzésből nyelvi adatbázist készítettek [7]. Az XML-fájl a lejegyzett beszélgetések elemzett változatát tartalmazza, amelyben megtalálhatóak a következő információk: egyértelműsített morfológiai elemzés, szótő; a regularizált szótő CV váza és a magánhangzók BNF (back, neutral, front) alakban; valamint az elhangzott szóalak fonetikai reprezentációja.

Az adatvédelmi törvény legfontosabb előírásai: megszüntetni az adatok és az érintettek közti kapcsolatot az érzékeny adatok törlésével az átiratokban, illetve kisípolásával a hangfelvételekben, emellett az egész hangfelvétel eltorzítása. Emellett szem előtt tartottuk azt is, hogy a BUSZI szociolingvisztikai interjúként csak akkor tud jól

funkcionálni, ha a beszélők egyes kutatási szempontból szükséges jellemzőit meghagyjuk – ilyen fontos jellemző például a születési hely, illetve az a hely, ahol az adatközlő a gyermekkorát töltötte. Az ilyen, szociolingvisztikai szempontból kiemelkedően fontos adatokat tehát meghagytuk, ám a beszélő anonimitását így is igyekeztünk biztosítani. Az a cél vezérelt bennünket, hogy minden tulajdonnév mechanikus törlése, illetve kisípolása helyett csak annyi, az adatközlőre utaló információt töröljünk, amelyek meghagyása veszélyeztette volna az adatközlő anonimitását.

A BUSZI-2 korpusz minden interjúja egyedi, ezért nem alkalmaztunk egységes szabályokat az anonimizálás során, hanem minden interjú adatait egyenként szűrtünk. A tesztfeladatok esetében csak a hangfelvételek torzítására volt szükség. Ehhez a SoX 14.4.0 programot használtuk.[4] A torzítás mértékét minden esetben az adatközlő hangjához igazítottuk. Az irányított beszélgetésekben minden esetben töröltük az adatközlő nevét és születési dátumát, és jellemzően töröltük a következő adatokat: munkahely címe, adatközlő jelenlegi és korábbi iskoláinak neve, az adatközlő lakhelye (utca). Nem töröltük viszont általában a következőket: születési hely, az adatközlő rokonainak adatai (kivétel a teljes név), az adatközlő lakhelye (kerület/városrész).

A doc formátumú átiratokban az érzékeny adatok helyén egy címke szerepel a törölt adat jellegével (pl. utcanév, személynév), az XML-formátumban MASKED címke szerepel a törölt elem helyén. A hangfájlokban a törölt elemek helyére azonos hosszúságú sinusjelet szúrtunk be, végül a teljes felvételt torzítottuk.

Egy példa anonimizált adatra a doc fájlban:

tm: december kilenc ■ öö kollégium a színhelye ■ az interjúnak. Budapesti interjú kvóta. ■ Be akarja mondani a nevét mar= vagy a marad a ka= marad inkább névtelenül?

ak: Jó mondom, mondhatom.

tm: Akko tessék.

ak: [#szemelynev]nak hívnak.

4 A BUSZI-2 kutathatóságáról

A BUSZI-2 ma már regisztrált kutatók számára hozzáférhető az interneten.[5] A tesztfeladatok szabadon elérhető eredményeinek feldolgozását egy keresőprogram segíti, melynek segítségével kilistázhatjuk, illetve meg is hallgathatjuk a részleteket (torzított formában).[6] Az irányított beszélgetések anonimizált átiratait doc, pdf és XML-formátumban tölthetik le a kutatók. Az ötven interjú átiratainak kutatását is egy keresőprogram segíti.[7] Az anonimizált és torzított hangfájlokat (a NECTÉ-hez és BNC-hez hasonlóan) nem tettük interneten elérhetővé, azok meghallgatására a Nyelvtudományi Intézetben előzetes időpont-egyeztetés után van lehetőség.

[4] http://sox.sourceforge.net/

[5] Részletesen lásd: http://buszi.nytud.hu/kutatni-szeretnem-a-buszi-t

[6] A tesztfeladatok keresőprogramját Blága Szabolcs készítette.

[7] Az irányított beszélgetések keresőprogramját Sass Bálint készítette.

Bibliográfia

1. Adolphs, S., Dawn, K.: Building a spoken corpus: What are the basics? In: O'Keeffe, A., McCarthy, M. (eds.): The routledge handbook of corpus lingusitics. Routledge, London (2010) 38–52
2. Beal, J. C.: Creating corpora from spoken legacy material. In: Renouf, A., Kehoe, A. (eds.): Corpus linguistics: Refinements and reassessments. Rodopi B. V., Amsterdam (2009) 33–47
3. Burnard, L.: A Note on Anonymization. Elérhető: http://www.natcorp.ox.ac.uk/archive/vault/pcw47.txt
4. Hunston, S.: Collection strategies and design decisions. In: Lüdeling, A., Merja, K. (eds.): Corpus linguistics: An international handbook. Volume 1. Mouton de Gruyter, Berlin (2008) 154–167
5. Kontra M.: Budapesti élőnyelvi kutatások. Magyar Tudomány, Vol. 5 (1990) 512–520
6. Labov, W.: Field methods of the project on linguistic change and variation. In: Baugh, J., Sherzer, J. (eds.): Language in use: Readings in sociolinguistics. Prentice-Hall, Englewood Cliffs, N. J. (1984) 28–53
7. Oravecz Cs., Sass B.: Szöveges lejegyzésből nyelvi adatbázis. Előadás. Elhangzott: BUSZI I. szimpózium, Budapest, 2008. december 9. Elérhető: http://www.nytud.hu/oszt/korpusz/resources/ocs_sb_buszidb.pdf
8. Váradi T.: A Budapesti Szociolingvisztikai Interjú. In: Kiefer, F. (szerk.): A magyar nyelv kézikönyve. Akadémiai Kiadó, Budapest (2003) 339–360

OpinHuBank: szabadon hozzáférhető annotált korpusz magyar nyelvű véleményelemzéshez

Miháltz Márton

MTA Nyelvtudományi Intézet
mmihaltz@gmail.com

Az előadásban szeretnénk bemutatni az OpinHuBank véleményannotált korpuszt, melyet számítógépes érzelem-/véleményelemző rendszerek kutatásához, tanításához és teszteléséhez fejlesztettünk ki. A korpusz a META-SHARE[1] disztribúciós hálózaton keresztül szabadon hozzáférhető.

A számítógépes véleményelemzés (opinion mining) célja a szövegekben megjelenő szubjektív kifejezések – érzelmek, értékelések, álláspontok, vélemények, hiedelmek, gondolatok, érzések, ítéletek, spekulációk stb. – és azok polaritásának (pozitív vagy negatív), valamint célpontjának (melyik, a szövegben megnevezett entitásra irányul) feltárása [1]. A technológia a weben napi szinten elérhető milliós nagyságrendű szöveges forrás (blogok, fórumok, közösségi oldalak, hírportálok) felhasználásával olyan alkalmazási területeket tesz lehetővé, mint az üzleti döntések támogatása, brandek monitorozása, piaci elemzések, online közvélemény-kutatások stb.

Magyar nyelven Berend és Farkas a kettős állampolgárság témájában megnyilvánuló hozzászólók véleményének automatikus megállapítását tűzte ki célul gépi tanulásos megoldással [2]. Az OpinHu projekt (2009-2010) [3] a magyar mellett angol, német, kínai és arab nyelven működő, szabályalapú tartalomelemző rendszert fejlesztett ki, melyben érzelmi szótárként a Harvard General Inquirer lexikon lokalizációját (magyarul mintegy 4700 kifejezés), polaritást módosító mintákat (magyarul kb. 30 elem), valamint mély nyelvi elemzésre támaszkodó mintafelismerést alkalmaztak.

Az OpinHuBank projekt egy olyan kézzel annotált, magyar nyelvű erőforrás létrehozását tűzte ki célul, mely megfelel az alábbi célkitűzéseknek (hasonlóan [4]-hez):

- A korpusz mérete tegye lehetővé gépi tanuló rendszerek betanítását is (10,000 különböző annotált példa kontextussal együtt.)
- A korpusz nyelvezete a magyar híroldalak és blogok szövegét reprezentálja. A kész korpusz anyagának 27%-a blogokból, 73%-a hírportálok, hírügynökségek oldalairól származik
- A vélemények polaritása mellett a vélemények célpontjai is annotálva vannak, hogy a korpusz segítségével célpontokat detektáló módszereket is lehessen vizsgálni, fejleszteni. A célpontok a korpuszban névelemek (tulajdonnevek).

1 http://www.meta-share.eu

- Minden annotációs egységet több, egymástól független humán annotátor is lásson el jelöléssel (5 különböző annotátor).

A korpusz anyagának gyűjtéséhez az OpinHu projekt[2] adatbázisa szolgálta a kiindulópontot, mely több mint 500 meghatározó, rendszeresen frissülő hazai online forrásból (híroldalak, blogok, fórumok) több millió különböző szöveget tartalmaz a 2009-2012 közötti időszakból. Az annotációhoz előkészítést az alábbi lépésekben végeztük el.

Első lépésben a szövegeket automatikus mondathatár-felismerő, tokenizáló (hontoken), morfológiai elemző (hunmorph) és szófaji egyértelműsítő (hunpos), majd névelem-felismerő (huntag) eszközökkel dolgoztuk fel. A teljes adatbázis összes cikkének minden mondatából véletlenszerűen kiválasztottunk 12,000 különböző mondatot, amely tartalmazott legalább egy, PERSON (személynév) típusú entitást, valamint megfelelt néhány biztonsági kritériumnak (legalább 7 token hosszú, a végén található írásjel). Minden mondat minden különböző entitás-előfordulása egy-egy különböző annotációs egység lett (így, ha ugyanaz a név ugyanabban a mondatban többször is előfordul, akár mindegyik előfordulás különböző polaritásannotációt kaphat). Mivel az entitásokat elsősorban vélemények célpontjaként szerettük volna felhasználni, kiszűrtük azokat a példákat, ahol az entitás nagy valószínűséggel a mondatban megjelenő vélemény forrása volt. A leggyakoribb ilyen szerkezetek a mondaton belüli – egy szónál hosszabb – idézetek, az ilyet tartalmazó mondatokban az idézőjelen kívül (a főmondatban) előforduló neveket nem soroltuk az annotálandó egységek közé. Végül kézi ellenőrzéssel kiszűrtük azokat a példákat, amelyekben az entitásfelismerő hibázott (kategóriatévesztés), az így fennmaradó annotációs egységek közül kiválasztottuk az első 10,000 darabot.

Az annotációhoz a GeoX Kft. munkatársai létrehoztak egy webes felületet, ahol az annotátorok saját azonosítóikkal belépve, annotációs egységenként haladva végezhették el a munkát. Az 5 annotátor számára megfogalmazott irányelvben a következőket rögzítettük: a polaritások megítélésében csakis az entitás adott mondatban megfigyelhető státuszát vegyék figyelembe, ne a névhez kapcsolódó elsődleges szubjektív asszociációjukat, világról való tudásukat (ehhez segítség: képzeljék el, hogy a mondatban az entitás helyén az ő nevük szerepel, hogyan tetszene így a mondat?); ne a teljes mondat polaritását vegyék figyelembe, ez lehet különböző a benne szereplő kérdéses entitás polaritásától; ha egy mondatban egy névhez pozitív és negatív vélemény is kapcsolódik egyszerre, semleges polaritást jelöljenek; ha egy entitás polaritása nem dönthető el rövid gondolkodás után/bizonytalan, legyen semleges a polaritása.

Az elkészült, szabadon hozzáférhető korpusz CSV fájl formátumban tartalmazza az annotációs egységeket, melyek az alábbi elemekből állnak: az egység azonosítója, a mondat azonosítója, a mondatot tartalmazó szöveg eredeti URL-je, a mondat szövege, a célpont entitás szövege, a célpont entitás kezdőpozíciója és hossza a mondatban (tokenek), az 5 annotátor jelölései (-1: negatív, +1: pozitív, 0: semleges).

Az OpinHuBank munkálatai a CESAR projekt[3] támogatásával készültek el.

2 https://sites.google.com/a/geox.hu/opinhu/

3 http://cesar.nytud.hu/

Hivatkozások

1. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proceedings of HLT/EMNLP 2005 (2005)
2. Berend, G., Farkas, R.: Opinion Mining in Hungarian based on textual and graphical clues. In: Proceedings of the 4th Intern. Symposium on Data Mining and Intelligent Information Processing. Santander (2008)
3. Miháltz M.: OpinHu: online szövegek többnyelvű véleményelemzése. In: VII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged (2010) 14–23
4. Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., van der Goot, E., Halkia, M., Pouliquen, B., Belyaeva, J.: Sentiment Analysis in the News. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'2010). Valletta, Malta (2010) 2216–2220

Miből lesz a robot MÁV-pénztáros?

Nemeskey Dávid, Recski Gábor, Zséder Attila

MTA SZTAKI
Nyelvtechnológiai Kutatócsoport
e-mail: ndavid,recski,zseder@sztaki.hu

A MÁV-pénztáros demonstrációban a felhasználó különféle vonatjegyeket vagy menetrendi információkat kérhet a programtól természetes nyelven. A rendszer két fő összetevőből áll: az egyik egy keretrendszer, amely lehetővé teszi, hogy a rendszer különböző komponensei akár más-más gépeken, aszinkron módon fussanak, míg a másik a tényleges szemantikai kód. A teljes program pythonban íródott, viszont az egyes komponensek bármilyen nyelvűek lehetnek.

A keretrendszer egy egyszerű eseményvezérelt architektúrát valósít meg, amelybe tetszőleges funkciójú összetevőket (*plugin*-eket) kapcsolhatunk be. Ezektől a komponensektől csak annyit követelünk meg, hogy a más komponensektől a keretrendszeren át érkező üzenetekre – melyek bármilyen python objektumot tartalmazhatnak – valamely más komponens számára értelmes választ adjanak. A keretrendszer segítségével rugalmasan működhetnek együtt a különböző elemzők, következtetőrendszerek vagy bármilyen külső erőforrás.

A tényleges nyelvi megértést végző szemantikai modul is egy a keretrendszerbe kapcsolódó plugin-ek közül. A `MachineCore` nevű komponens gondoskodik a felhasználói üzenetek belső reprezentálásáról, a háttértudások tudásbázisba építéséről, a mondatok szintaktikai és szemantikai elemzéséről és a következtetésről. A felhasználótól érkező szöveges (nem hangalapú) üzenetek feldolgozásához a szemantikai magnak morfológiailag elemzett és fő mondattani összetevőkre bontott adatokra van szüksége. Ezeket az adatokat a `hunmorph` [1] és a `hunchunk` [2] eszközökkel nyerjük ki. A jelen alkalmazás céljaira ezután egy egyszerű, mindössze néhány reguláris kifejezésből álló dátum- és időpontfelismerő komponenst is lefuttatunk.

Az így elkészült adat ezután készen áll a szemantikai feldolgozásra. Az elemzés és következtetés motorja a *Spreading Activation (SA)*, melynek alapegységei az ún. *gépek* (machine, definícióját l. [3], 10. fejezet), majd a külvilág, pontosabbab bizonyos, külvilággal érintkező plugin-ek felé az eredményeket attribútum-érték mátrixokon (AVM) keresztül kommunikálja.

Az egyes szavak jelentésének leírására a kutatócsoport kidolgozott egy definíciós szintaxist [4] és fejlesztett egy parszert, mely ezt a tudást ugyanabban a gépalapú reprezentációban ábrázolja, amellyel a rendszerünk dolgozik. Így egyszerű szöveges fájlokkal írhatjuk le a fogalmak jelentését. Ez a modul lehetőséget teremt arra is, hogy a szavak fogalmi szintű leírásához különböző nyelvű szavak, kifejezések is kapcsolódhassanak. A demóhoz létrehoztunk egy MÁV-jegyekkel és vonatokkal kapcsolatos rövid leíró fájlt, amely csak a témakörrel kapcsolatos szavakat definiálja.

Az SA a konstrukciós nyelvtanból ismert konstrukciókat futtatja le az előfeldolgozott szövegen – a szavak között található kapcsolatokat ily módon ismerjük fel. Ilyen konstrukció például a főnévi csoportokon belüli jelzők főnévhez való kapcsolása vagy az igei vonzatkeretek kitöltése. Ez utóbbihoz még arra is szükség volt, hogy az igék definiálásakor helyet hagyjunk az egyes vonzatoknak, és egy külön fájlban írjuk le, hogy mely vonzatok általában milyen szerepet töltenek be a mondatban.

Az SA másik fontos funkciója, hogy teljes futása során számontartja egy ún. *aktív tömbben*, mely szavakkal, illetve gépekkel dolgozunk. Így ha olyan szavakat lát az aktív gépek között, melyeknek ismerjük a jelentését (mivel definiáltuk őket), akkor a jelentésüket is beírjuk ebbe az aktív tömbbe, vagy fordítva: ha úgy látja, hogy már létrejött egy gépekből álló struktúra, amely épp egy szó jelentésének felel meg, akkor a szót is felvesszük. Pl. a *menetrend* és a *vonat* szavak a mi rendszerünkben életre hívják az *elvira* gépet, vagy amikor a rendszer meglátja a *megy* igét, akkor helyettesíti azzal a struktúrával, amely leírja, hogy ez az ige mit jelent valójában.

Végül az SA gondoskodik arról is, hogy az AVM-ek a futás során kitöltődjenek azokkal az értékekkel, amelyeket a külső erőforrások megkövetelnek. Egyszerre több AVM is kitöltődhet, mert a mondat egyes szavaiból még nem tudhatjuk, hogy a kér(d)és mire vonatkozott, így arra is fel kell készülni, hogy a felhasználó menetjegyet vagy helyjegyet kért, esetleg mindkettőt, de arra is, hogy valójában a menetrendről érdeklődött. Az SA során létrejövő aktivációk sorozatából, a gépek és AVM-ek versengéséből valamelyik (esetleg több) győztesen kerül ki, és az kerül eredményként a felhasználó elé. Ez jelen esetben egy vagy két nyomtatott jegyet, esetleg egy külön böngészőablakban megjelenített menetrendi információt jelent.

Hivatkozások

1. Trón, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: open source word analysis. In Jansche, M., ed.: Proc. ACL 2005 Software Workshop. ACL, Ann Arbor (2005) 77–85
2. Recski, G., Varga, D., Zséder, A., Kornai, A.: Főnévi csoportok azonosítása magyar-angol párhuzamos korpuszban. In: VI. Magyar Számitógépes Nyelvészeti Konferencia. (2009)
3. Eilenberg, S.: Automata, Languages, and Machines. Volume A. Academic Press (1974)
4. Kornai, A., Makrai, M.: A 4lang fogalmi szótár. In: IX. Magyar Számitógépes Nyelvészeti Konferencia. (2013)

Az új magyar Braille-rövidírás korpuszvezérelt kialakításának lehetőségei

Sass Bálint

MTA Nyelvtudományi Intézet
sass.balint@nytud.mta.hu

A vakok által világszerte használt, tapintáson alapuló Braille-írást Louis Braille fejlesztette ki 1837-ben. A karakterek (ún. Braille-cellák) két oszlopban elrendezett 3-3, azaz összesen hat kidomborodó pontból állnak. Az egyes pontokra a sorszámukkal hivatkozunk: a bal oszlopban helyezkedik el fentről lefelé az 1-es, 2-es, 3-as pont, a jobb oszlopban pedig szintén fentről lefelé a 4-es, az 5-ös és a 6-os. A kidomborodó és ki nem domborodó pontok különböző elrendezéseiből összesen $2^6 = 64$ különböző karakter áll elő: a *t* jele például a 2345 (⠞).

Egyrészt mivel a Braille-írás papíron (speciális dombornyomtatóval kinyomtatva) sok helyet foglal el, másrészt az írás (jegyzetelés) és az olvasás meggyorsítására kialakították a Braille-rövidírásokat – külön-külön az egyes nyelvekre (német: [1]; angol: [2]). A rövidírásban szabályok adják meg, hogy mit hogyan rövidítünk. Magyarban az 50-es években kidolgozott és ma is használatos ún. „kis" rövidírásban például a *hogy* szót *h*-val (125 ⠓), a *kell*-t *k*-val (13 ⠅) rövidítik.

A Magyar Vakok és Gyengénlátók Országos Szövetsége 60 év elteltével döntött úgy, hogy a mai nyelvhasználatot is figyelembe vevő új rövidítésekkel bővíti a szabályrendszert, azzal a céllal, hogy a rövidítési képessége a jelenlegi nagyjából 10%-ról a 20% közelébe növekedjen. A rövidírás-rendszerek kifejlesztése sok esetben nagy időigényű feladat, az egységes angol rövidírás kialakítása 1991-től kezdődően majdnem két évtizedet vett igénybe [3].

Jelen kutatásban azt vizsgáljuk, hogy hogyan lehet korpuszgyakorisági adatok alapján, a lehető legkisebb emberi beavatkozással, azaz szinte teljesen automatikusan és ezáltal záros időn belül előállítani a lehető legnagyobb rövidítési képességgel bíró új magyar rövidírást. Nyilván a lehető leggyakoribb elemeket (karaktersorozatokat) érdemes a lehető legrövidebbre rövidíteni, így nyerjük összességében a legtöbbet. A szűk keresztmetszet a rövidítésre rendelkezésre álló jelek száma: a 64 egykarakteres jelből is csak a ritkábbak alkalmasak arra, hogy rövidítésjelek legyenek.

Minden karaktersorozathoz ötféle gyakorisági értéket rendelünk: hányszor fordul elő (1) szó elején, (2) szó belsejében, (3) szó végén, (4) önálló szóként, illetve az előző négy összegeként: (5) összesen. Ez az elkülönítés két okból is kiemelten fontos. Egyrészt adott jelet rövidítésként csak abban a pozícióban érdemes szerepeltetni, ahol nem (vagy alig) fordul elő (például: pontosvessző szó elején, szó belsejében vagy önálló szóként). Másrészt pedig, azáltal, hogy nem

csak összesített gyakoriságokkal dolgozunk, megmarad annak a lehetősége, hogy egy adott rövidítésjelet különféle pozíciókban eltérő célokra használhassunk, ahogy például az `1346` (⁘) a németben a szó belsejében lévő *mm* rövidítése és az *immer* önálló szó rövidítése is. Utóbbi esetben amiatt szabadul fel egy rövidítésjel önálló szó rövidítésére, mert tekintetbe vették, hogy az *mm* betűkapcsolat a németben önálló szóként nem fordul elő. Ugyanezt az elvet követhetjük a magyarban is: a nagyon gyakori *et* hangkapcsolatra alkalmazott rövidítésjelet önálló szóként a *szerint* rövidítésére használhatjuk, mivel az *et* önálló szóként extrém ritka.

Az algoritmus vázlata a következő. Számba vesszük a rövidíthető nyelvi elemeket, azaz a gyakori karaktersorozatokat. Egy gyakorisági listában tüntetjük fel mind az öt típust a fenti ötféle gyakorisági értékükkel külön-külön, így egy elem 5-ször fog szerepelni. A listát a várható rövidítési képesség szerint rendezzük. A rövidítési képességet a következőképpen számoljuk: $\mathrm{rk}(w, r(w)) = [l(w) - l(r(w))] * fq(w)$, ahol w az eredeti rövidítendő karaktersorozat, $r(w)$ a rövidítés, $l()$ a hossz (karakterszám), $fq()$ a gyakoriság. Először abból indulunk ki, hogy 1 hosszúságú rövidítéseket tudunk képezni. A legritkábban előforduló jelek lesznek alkalmasak rövidítésnek. Ismerve ezek listáját, hozzárendeljük a lista első helyén álló rövidítendő elemhez a legritkább elemet rövidítésként.

A szabályok automatikus megalkotásakor számos szempont figyelembevételével döntünk. Az egyes rövidítésjelek hatékony felhasználására vonatkozó fenti megfontolások szerint járunk el. Kezeljük azt az esetet, mikor a rövidítésjel literálisan jelenik meg (azaz például az `125` (⁖) nem rövidítésként, hanem valóban *h* betűként értendő – ilyenkor egy külön erre szolgáló jellel prefixáljuk az adott jelet, és ez levonódik a rövidítési képességéből). Egy előzetesen kidolgozott lista alapján bizonyos toldalékok (pl.: *-ság/-ség*) hangrend szerint eltérő formáit összevonjuk, egy jellel rövidítjük. Számításba vesszük a korábbi rövidítésekkel való átfedések hatását, ugyanis egy szabály megalkotása érinti az általa rövidített elem részleteire vonatkozó vagy az elemet részként tartalmazó potenciális szabályokat, az általuk elérhető rövidítési képesség változhat. Ezért minden szabály megalkotása után újrarendezzük a listát a frissített rövidítési képességek szerint, majd vesszük a lista elejére kerülő – legnagyobb rövidítési képességgel bíró – elemet, és visszatérünk az algoritmus elejére. Ha nem találunk megfelelő n hosszúságú rövidítésjelet az aktuális rövidítendő elemhez, akkor a továbbiakban $n + 1$ hosszúságú többkarakteres vagy speciális prefixjellel ellátott rövidítésjelet keresünk hozzá. Ennek megfelelően a fenti képlet szerint csökken az elemhez rendelt rövidítési képesség, és ez befolyásolja a rendezett listán elfoglalt helyét is.

A legnagyobb lehetséges rövidítésre vonatkozó fent tárgyalt megfontolások mellett ugyanilyen fontos szempont az új rövidírás jó olvashatósága (tapintás útján jó felismerhetőség) és könnyű megtanulhatósága (kevés, egyszerű szabály). A nagy rövidítési képesség és kényelmes használhatóság egymás ellen ható követelmények, itt egy körültekintően kidolgozott kompromisszumra valamint közvetlen vakok általi tesztelésre van szükség annak érdekében, hogy a potenciális

felhasználók elfogadják és szívesen használják az új rövidírást. A Bánó-féle ún. ,,nagy" rövidírás éppen bonyolultsága miatt nem terjedt el korábban.

Tapasztalat szerint a jól olvasható rövidítés pozíciótól függetlenül mindig azonos jelentésű, a szó kezdő és záró betűjéből, illetve a szót alkotó jellegzetes mássalhangzóból áll. Érdemes külön kezelni az egykarakteres és a többkarakteres rövidítéseket ebből a szempontból. Az egykarakteres rövidítésjelek kiemelten értékesek, mivel nagyon rövidek és nagyon kevés van belőlük. Esetükben nyilván nem követelhető meg a szó kezdő és záró betűjére vonatkozó fenti feltétel, főként, hogy a legalkalmasabb rövidítésjelek éppen az írásjelek. Érdemes megengedni, hogy az egykarakteres rövidítésjelek esetében csak a rövidítési képesség számítson, azaz korlátozás nélkül bárminek a rövidítésére felhasználhassuk őket, sőt még azt is, hogy különböző pozíciókban különféle jelentéssel bírhassanak. Lehetőség szerint törekedni kell a könnyű megtanulhatóságra, ahogy ezt a fent idézett német (*mm*) és magyar (*et*) példánál láttuk. A többkarakteres rövidítésjeleknél a fenti követelmény könnyebben teljesíthető, esetleg automatikus úton is.

Említettük, hogy a lista elején aktuálisan található leggyakoribb rövidítendő elemhez mindig az épp rendelkezésre álló legritkább elemet rendeljük hozzá rövidítésként. A fentiek alapján ez egykarakteres rövidítés esetén valóban szigorúan gyakorisági alapon történik a lehető legnagyobb rövidítés elérése érdekében. Az olvashatósági szempontok akkor kerülnek előtérbe, mikor a rövidítendő elem ritka típusa helyett keresünk másik ideillő, könnyen megjegyezhető rövidíthető elemet; illetve a többkarakteres rövidítéseknél, mikor számos azonos gyakoriságú (ritka) rövidítésjel közül választhatunk.

A fenti módszerrel előállított rendszer rövidítési képessége elérheti a kívánt 18-20%-ot, ami megfelel az új magyar Braille-rövidírással szemben támasztott követelményeknek.

Hivatkozások

1. Freud, E.: Leitfaden der deutschen Blindenkurzschrift: Teil 2. Verlag der Deutschen Blindenstudienanstalt, Marburg (1973)
2. Christine Simpson, ed.: The Rules of Unified English Braille. Version I edn. Round Table on Information Access for People with Print Disabilities Inc., Australia (2010)
3. Bogart, D.: Unifying the English Braille Code. Journal of Visual Impairment & Blindness **103**(10) (2009) 581–583

Neticle – Megmutatjuk, mit gondol a web

Szekeres Péter

Neticle Technologies Kft. (http://www.neticle.hu)
Budapest, Magyarország
peter.szekeres@neticle.hu

Kivonat: A Neticle rendszer (http://www.neticle.hu/) számokban foglalja össze a web véleményét egy adott témával, ha úgy tetszik, kulcsszóval kapcsolatban. Ehhez az egyik legfontosabb mutatónk az úgynevezett webes véleményárfolyam, mellyel egyszerűen követhető a webes jelenlét.

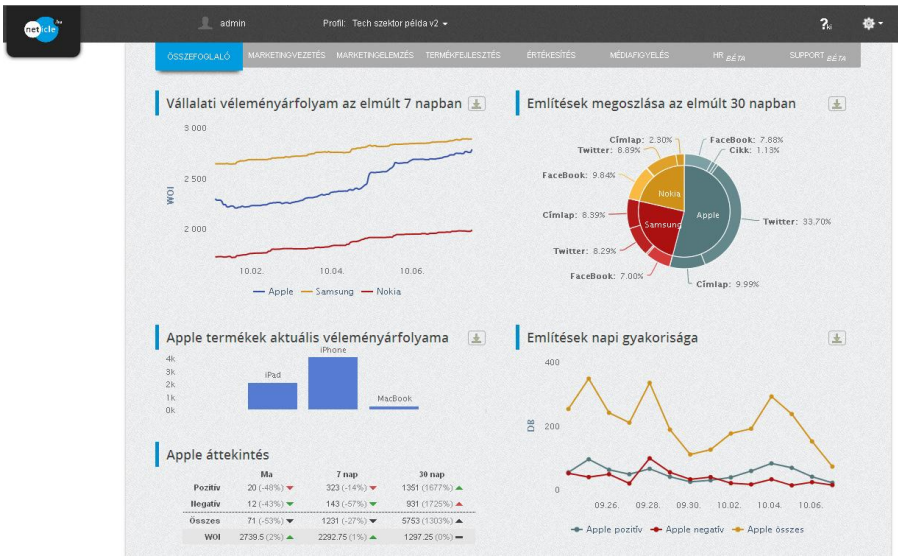
1 Bevezetés

Manapság rengeteg írás keletkezik a weben: az internezők értékelnek, támogatnak, beszámolnak, kritizálnak. Terméket, szolgáltatást, céget, piacot, eseményt. Sok hasznos információ és tudás hever szerte a világhálón.

A Neticle-lel az volt a célunk, hogy egy olyan webes szolgáltatást hozzunk létre, amelynek segítségével a lehető legegyszerűbben felhasználhatjuk ezeket az információkat a üzleti döntéseinkhez. A Neticle (http://www.neticle.hu) olyan böngészőből elérhető szolgáltatás, amely a magyar nyelvű webes szöveges információk automatikus elemzésével, értékelésével és vizualizálásával a közel valós idejű nyomon követést és a tényalapú döntéshozatalt támogatja.

2 Adatok gyűjtése

A rendszer alapja egy olyan crawler (kereső) elkészítése volt, mely a web magyar nyelvű tartalmait megkeresi és kategorizálja előre meghatározott csoportok szerint. Az így kialakított osztályok segítségével a crawler meghatározza, érdemes-e az oldalt újralátogatni, és ha igen, akkor milyen gyakran, hogy a frissített tartalmakat vagy bővülő hozzászólásokat minél hamarabb megtalálja a rendszer. A webes médiumtípusok automatikus meghatározása a rendszer által talált tartalmak későbbi felhasználását és feldolgozását is elősegíti [1]. A weboldalak mellett a fő közösségi oldalak (Twitter és Facebook) nyilvános posztjait is feldolgozzuk.



1. ábra. A Neticle rendszer egy képernyője.

3 Véleményárfolyam számítása

A Neticle a megtalált szövegek elemzésével számokban foglalja össze a web véleményét, hangulatát egy adott témával, ha úgy tetszik, kulcsszóval kapcsolatban. Ehhez az egyik legfontosabb mutatónk az úgynevezett webes véleményárfolyam

A Neticle véleményárfolyam (**WOI, Web Opinion Index**) egy univerzális mutató, amely összefoglalja a webes tartalmak véleményét egyetlen dinamikusan változó számban. (Ez a mutató tulajdonképpen a tőzsdei részvényárfolyam analógiája, a weben publikálók véleményét, hangulatát tükrözi.)

A pozitív és negatív webes tartalmak alapján számolt index egyértelműen mutatja egy cég/termék/téma webes megítélését illetve annak változását: a tartalmak polaritását számszerűsíti a rendszer, polaritásindexet rendel a szövegekhez. Így ha egy pozitív írás jelenik meg a témában a weben, akkor az árfolyam növekszik, ha pedig valaki negatívan nyilatkozik egy fórumon például a termékről, akkor a véleményárfolyam csökken. Az árfolyam összevethető múltbeli adatokkal és a versenytársak árfolyamaival, vizsgálhatóak a marketingkampányok hatásai is például.

A fejlesztés során az elképzelésünk az volt, hogy a magyar nyelvű webes mondatok véleménypolaritásának (tehát pozitív-negatív voltának) számítógépes meghatározása a megfelelő algoritmussal elérheti az emberi ítélőképesség határát. Azaz közel 82%-ban egyezhet egy ember által elvégzett manuális pozitív-semleges-negatív értékeléssel Az eddigi tesztek alapján különböző témakörökben 70-80%-os pontosságot sikerült elérni az automatikus polaritásmérő algoritmusunkkal [2].

4 Eredmények

Az automatikus polaritásméréssel megvalósított közel valós idejű webes véleményárfolyam számítással egyszerűen követhető a Neticle-ben, hogy mit gondol a web a cégünkről, termékünkről.

Hivatkozások

1. Csikós, Z., Szekeres, P.: Friss tartalmak gyors megtalálása a magyar weben. Budapesti Corvinus Egyetem Tudományos Diákköri dolgozat (2012)
2. Szekeres, P.: Polaritásmérés magyar nyelvű webes szövegekben. Budapesti Corvinus Egyetem, Budapest (2012)

Vektortér alapú szemantikai szóhasonlósági vizsgálatok

Tóth Ágoston

Debreceni Egyetem, Angol-Amerikai Intézet
4010 Debrecen, Pf. 73.
toth.agoston@arts.unideb.hu

Kivonat: A bemutatott kísérletben kiválasztott szavakat a környezetükben előforduló szavak gyakorisági adataiból képzett vektorokkal reprezentáljuk, a vektorok összehasonlításával pedig a szavak szemantikai hasonlóságára következtetünk. A kísérleti rendszer egy feleletválasztásos feladatot old meg, melyben 30 célszó mindegyikéhez automatikusan kiválasztjuk a hozzá leghasonlóbb szót. A vizsgálandó szavak listáján 15 szemantikailag motivált párt találunk, köztük ellentéteket, szinonimákat és alá-/fölérendelt szavakat; kimenetként mindegyik szó párját vártuk visszakapni. A helyes választ a rendszernek mind a 30 szóhoz összesen 100 potenciális jelölt közül kellett kiválasztania. A pontosság maximális értéke (20 millió szavas korpusz feldolgozása után) 79% volt. A vektorokat a Magyar Webkorpuszból vett, annotációt nem tartalmazó szövegek segítségével állítottam elő, a vektorok kiszámítását és összehasonlítását saját fejlesztésű programmal végeztem.

1 Bevezetés

A szavak előfordulási gyakoriságára vonatkozó megfigyelések az ember és a gép által is könnyen gyűjthető adatok, melyek önmagukban is megalapozzák szemantikai jellegű feladatok megoldását.

Az első vektortér alapú szemantikai eredmények az információkeresés területén születtek (l. pl. [7]). Egy dokumentum a benne előforduló szavak gyakorisági adataival jellemezhető, melyekből (dokumentumokra jellemző) vektorokat hozunk létre. Ezáltal egyrészt a dokumentumok egymással összehasonlíthatók, másrészt az információkereséshez használt aktuális keresőkifejezésből ugyanilyen módszerrel létrehozott szógyakorisági vektorral a már meglévő vektorokat összehasonlítva a releváns dokumentumok megtalálhatók.

Szintén konstruálható olyan rendszer, amely nem dokumentumok, hanem szavak hasonlóságának mérését teszi lehetővé (l. pl. [3] és [6]). Ebben az esetben kiválasztott célszavakat olyan vektorokkal reprezentálunk, amelyek a környezetükben előforduló szavak gyakoriságát tükrözik. Az így kapott környezetvektorok összehasonlításával (pl. távolságuk meghatározásával) mérjük a szavak hasonlóságát, amelyet – a disztribúciós hipotézis [8] értelmében – szemantikai megfigyelésnek tekintünk. A vektorokat a környezetszavak által meghatározott sokdimenziós térben egyszerűen összehasonlíthatjuk úgy, hogy az origóból a vektorok által kijelölt pontok távolságát mérjük, vagy a

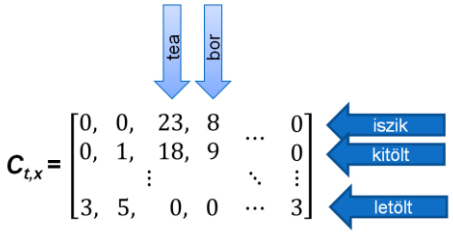
vektorok hajlásszögét állapítjuk meg. Az eljárás a szójelentés egy speciális közelítésének egyfajta geometriai modellezését jelenti.

Munkám egy olyan kísérletet mutat be, melyhez saját JAVA-alkalmazást fejlesztettem, mely nagyméretű korpuszokból automatikusan felépít előre meghatározott dimenzióval rendelkező vektortereket, és létrehozza a kijelölt szavakat jellemző vektorokat, amelyeket végül össze is hasonlít egy feladat megoldása során.

A cikk felépítése a következő: először bemutatom a szóhasonlósági kísérletemben használt rendszer felépítését a betanítás során használt paraméterek megadásával, majd leírom a kísérletben végrehajtott szemantikai feladatot, és értékelem a rendszer teljesítményét.

2 A kísérleti rendszer felépítése

Első lépésként egy mátrixot hozunk létre, melynek soraiban egy-egy *célszó* ábrázolását állítjuk elő (ezek megfelelnek a bevezetőben említett környezetvektoroknak), az oszlopok pedig egy-egy *környezetszónak* a célszavak környezetében megfigyelt előfordulási gyakoriságát reprezentálják. A mátrix egy eleme azt mutatja meg, hogy az adott célszó környezetében a feldolgozott korpuszban összesen hányszor fordul elő az adott pozícióhoz tartozó környezetszó.



1. ábra. Szó/környezet mátrix (t=*target*, x=*context*).

A mátrix sorait egy-egy környezetvektorként értelmezzük, amely a célszó és a környezetszavak közötti szintagmatikus kapcsolatokat ábrázolja. Például az 1-4 mondatok feldolgozása során az *ittam* szó környezetvektorában – egy 1+1 szavas szimmetrikus mozgó ablakot használva a környezet megfigyelésére – növelni fogjuk a következő szavaknak megfelelő vektorelemek értékét: *szóval, kávét, nem, teát* és *a*. Nagyobb, 2+2 szavas ablak esetén az *ittam* szó környezetvektorát befolyásolni fogják a *borból* és a *sörömet* szavak is. A vektorelem értéke arányos a célszó és az adott vektorelemnek megfelelő környezetszó közös előfordulásainak számával.

1. Szóval *ittam* kávét.
2. Nem *ittam* teát.
3. *Ittam* a borból.
4. *Ittam* a sörömet.

A környezetvektorok ábrázolásához szükséges vektorterek általában nagyon sok dimenzióval rendelkeznek, hiszen alapesetben minden, a jellemzett szavak környezetében előforduló környezetszó növeli a vektortér dimenzióját, amit utólag csökkenthetünk kezelhető méretűre. Jelen kísérletsorban elkerülöm a dimenzióredukciót azzal, hogy kizárólag a leggyakoribb (8-14 ezer) szót veszem figyelembe az ábrázolandó célszavak környezetében, a vektorok összehasonlítását pedig olyan egyszerű eszközökkel végzem, ami ilyen dimenziószám esetén is jól használható és gyors.

A célszavakat jellemző környezetvektorokat nem „nyers" formában (frekvenciaadatokkal) használtam fel, hanem belőlük a cél- és környezetszavakra pozitív pontszerű kölcsönös információt (pPMI) számoltam [2], ezzel mérve a két szó együttes előfordulásának valószínűségét azok *külön* történő előfordulásához képest.

Végül a pPMI értékeket tartalmazó vektorok összehasonlítása során a hajlásszögükből (α) számolt $\cos \alpha$ értékkel kaptam meg a célszavak hasonlóságát (vö. [5]). Előfeltevésünk szerint ez szemantikailag interpretálható mérték.

A hasonló kísérletek egyik fontos és általában hosszas munkával kikísérletezhető momentuma a lehetséges paraméterek megfelelő beállítása. Számos ilyen paraméter létezik a fent leírt, kifejezetten a rendszer felépítésére vonatkozó paramétereken kívül is. Ebben a kísérletben annotáció nélküli korpuszt használtam, tokenizálás és lemmatizáció nélkül, stopszavak használatát mellőzve (tehát a funkciószavakra vonatkozó gyakorisági adatok is megjelentek a környezetvektorokban, ami a pPMI vektorok és a hajlásszög alapú összehasonlítás miatt elvileg nyereséges döntés). A vektorok előállítása során a mozgóablak mérete 1+1 szó volt (bal és jobb oldalon 1-1 közvetlen szomszéd). Elsődleges célom a paraméterek beállítása során az angol nyelvre vonatkozó szakirodalmi adatok alkalmazhatóságának (elsősorban [1] alapján) kipróbálása volt a magyar nyelv feldolgozásában. Ebben a konkrét kísérletben a magyar és az angol nyelv közötti különbségek (gondolva itt elsősorban a nagyon különböző alaktani alrendszerekre) nem jelentettek problémát; ezzel együtt, bizonyos paraméterek eltérő beállításának a vizsgálata (pl. lemmatizáció használata) a későbbiekben szükséges lehet.

3 A szemantikai feladat, a rendszer pontossága

A vektortér alapú szemantikai rendszer tesztelésének módszertana egy további fontos kérdés, amire a nemzetközi szakirodalomban legalább 4 különböző eljárást találunk [1]:

- „TOEFL-teszt": feleletválasztós teszt, melyben néhány alternatíva közül kell automatikusan kiválasztani a megadott szóhoz jelentésben legközelebb állót;
- távolság összehasonlítása: ez is egy feleletválasztásos feladat, melyben adott célszavakhoz automatikusan kiválasztjuk a hozzá legközelebb álló szót; a választási lehetőségek tartalmaznak véletlenszerűen kijelölt szavakat a célszavak közül, valamint a vizsgált célszó egy előre kijelölt és célszavak közé felvett szemantikai párját (pl. szinonimáját, ellentétét, stb.), amit helyes kimenetként várunk;

- szemantikai osztályozás (előre kijelölt kategóriákba, pl. gyümölcsök, fegyverek, stb.);
- szófaji és mondattani klaszterezés.

Az itt bemutatott kísérleti rendszerben megoldandó feladatként egy távolság-összehasonlítási vizsgálatot választottam, amihez 30 célszót használtam. Ezek 15 szemantikailag motivált párt alkottak: voltak köztük szinonimák (pl. *egész–teljes*, *fut–rohan*, *néz–figyel*), ellentétek (*fekete–fehér*, *régi–új*, *ki–be*) és hiponimák/hiperonimák (alá-/fölérendelt szavak, avagy specifikusabb/általánosabb szavak, pl. *alma–gyümölcs*, *labdarúgás–sport*, *szekrény–bútor, kutya–állat*), egyforma számban. A figyelt szavak ilyen megadása azt biztosította, hogy mindegyik szóhoz volt egy „legközelebbi szó”, amely a rendszer által visszaadandó elvárt kimenet volt. A szavak kiválasztásánál a szófaji változatosságról gondoskodtam.

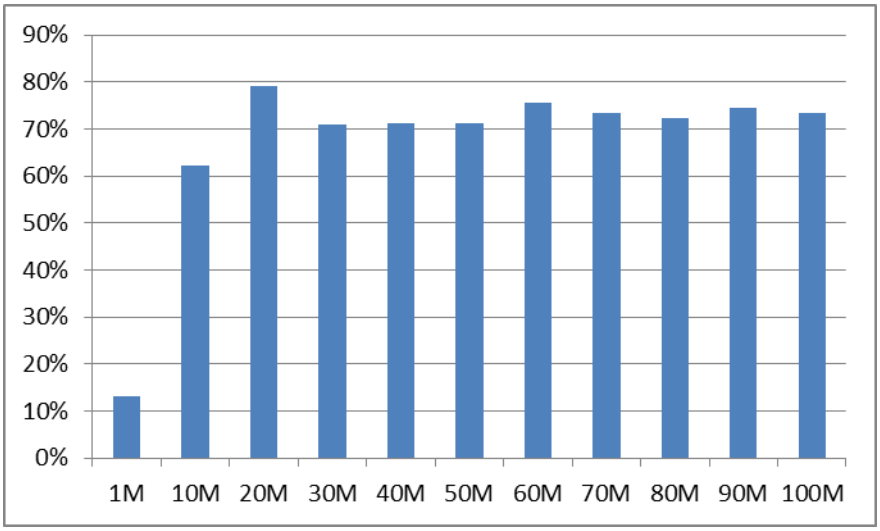
1. táblázat: A kísérlethez kiválasztott szavak.

Célszó	***Várt kimenet***
fekete	*fehér*
fehér	*fekete*
régi	*új*
új	*régi*
fent	*lent*
lent	*fent*
ki	*be*
be	*ki*
jó	*rossz*
rossz	*jó*
legmagasabb	*legnagyobb*
legnagyobb	*legmagasabb*
egész	*teljes*
teljes	*egész*
tép	*szakít*
szakít	*tép*
néz	*figyel*
figyel	*néz*
fut	*rohan*
rohan	*fut*
alma	*gyümölcs*
gyümölcs	*alma*
szekrény	*bútor*
bútor	*szekrény*
kutya	*állat*
állat	*kutya*
labdarúgás	*sport*
sport	*labdarúgás*
dollár	*deviza*
deviza	*dollár*

A helyes kimenetet a rendszernek mind a 30 szóhoz összesen *100 potenciális jelölt közül kellett kiválasztania*: a 100 alternatíva tartalmazta az eleve vizsgált 30 szót, valamint 70 olyan szót, amit a Magyar Webkorpusz [4] első 1000 leggyakoribb szavából választott a program véletlenszerűen. (Ilyen módon előfordulhat, hogy az opciók közé bekerül egy vagy több olyan szó, amely egy célszóhoz szemantikailag kapcsolódik. Ezt kizárni nem tudtam, de lent megadom a rendszer pontosságát arra az esetre is, amikor a 70 véletlenszerűen kiválasztott szó nem szerepelt a választható alternatívák között.) A véletlen elem miatt a futtatást többször megismételtem, és az eredményeket átlagoltam. A környezetvektorok kiszámítására a Magyar Webkorpuszból vett 100 millió szavas (annotáció nélküli) részkorpuszt használtam.

A random baseline pontosság 1% volt. A fedést ebben a tesztelési módszertanban 100%-on tartjuk: a feleletválasztás kikényszerített jellegű.

A pontosság 1 millió szó feldolgozása után átlagosan 13% volt, de ekkor még volt olyan célszó a 30 közül, ami a rendszer által figyelt környezetszavak (a Webkorpusz 14000 leggyakoribb szava) mellett még egyáltalán nem fordult elő a korpuszban. 10 millió szó után a pontosság 62%, 20 millió szónál 79% volt (baseline: 1%); ezután már nem javult a pontosság, egészen 100 millió szóig vizsgálva. A feldolgozott szavak száma a 2. ábrán látható módon befolyásolta a pontosságot.



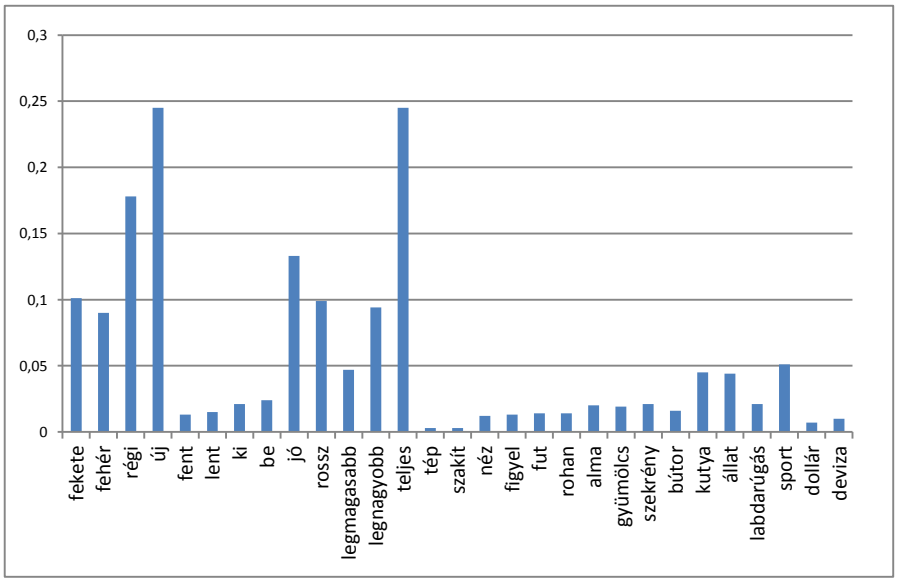
2. ábra. Pontosság változása a korpuszméret (millió szó) függvényében.

A választási lehetőségeknek a redukálása az eredeti 30 szóra javulást hozott (87% maximális pontosság 50 és 60 millió szavas korpuszméreteknél, 3%-os random baseline mellett). A választási lehetőségek 100-ról 250-re növelése a pontosságot csak enyhén, 77%-ra csökkentette (random baseline=0,4%).

A szakirodalomban elterjedt az, hogy a választási lehetőségek számát kifejezetten alacsony szinten tartják, így pl. 10 szóból választva (10% baseline mellett) elérhető 90% feletti pontosság is.

A kísérletbe bevont környezetszavak számát 8 és 14 ezer között vizsgáltam. Ennek a paraméternek a növelése marginális, de mérhető változást okozott (a környezetszavak számának emelése a pontosságot növelte, az elért növekedés néhány százalékos volt).

A számszerűsíthető eredmények mellett érdekes volt azon esetek vizsgálata, amikor egy adott szóhoz nem az elvárt kimenetet, hanem egy másik szót találtunk leghasonlóbbnak. A megfigyelt esetek egy része szemantikailag is értelmezhető volt. Ilyen például a *kutya→ember* és *állat→ember* asszociációk (*kutya↔ állat* helyett) abban az esetben, amikor a véletlenszerűen kiválasztott opciók között az *ember* szó is megjelent. Szintén a véletlen elemnek köszönhető probléma volt, amikor a *legmagasabb* szó párjának keresése közben a lehetséges válaszok közé bekerülő *magas* szó elnyomta az előre kijelölt párt (*legnagyobb*), ami tulajdonképpen nem is hiba, azonban az itt alkalmazott kiértékelési módszertanban a pontosság csökkenéséhez vezet. Természetesen arra is volt példa, hogy az algoritmus által visszaadott asszociáció szemantikailag motiválatlannak tűnő zaj volt, pl. *egész→új* megfeleltetés a kimenetként remélt *egész→teljes* helyett. A 3. ábra ezt az esetet mutatja be, szemléltetve az *egész* szó hasonlóságát az első 30 célszóhoz (a véletlenszerűen választott 70 szó hasonlósági értékeit itt helyhiány miatt nem ábrázoltam).



3. ábra. Célszavak hasonlósága az *egész* szóhoz (1=maximális hasonlóság).

Szemantikailag nem értelmezhető zaj esetén általánosnak volt mondható a 3. ábrán látható jelenség: a várt kimenetnek (ebben az esetben: a *teljes* szónak) és a zajnak (itt: *új*) a célszótól (*egész*) való távolsága nagyon hasonló volt, a harmadik, negyedik stb. helyezett szó jócskán lemaradva követte őket.

Általános tendenciaként megfigyelhető volt, hogy az alá-/fölérendelt szavaknál volt a legnagyobb a pontosság, ettől elmaradt az ellentétek és a szinonimák kezelése.

Munkám távlati célja a vektortér alapú számítógépes nyelvészeti megközelítés szisztematikus szemantikai vizsgálata, hiszen – miközben alkalmazásokban már megjelentek ezek az eszközök, és a velük kapcsolatos tapasztalatok egyre gyűlnek –, lexikai szemantikai szempontból az ilyen eljárásokat nem értékelték még mélyrehatóan. A számítógépes eszköz kifejlesztése és kipróbálása az itt bemutatott módon az ehhez szükséges első lépés volt.

Köszönetnyilvánítás

A cikk elkészítését részben az OTKA K 72983 számú kutatási projekt, részben pedig a TÁMOP 4.2.1./B-09/1/KONV-2010-0007 számú projekt támogatta. A TÁMOP projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Bibliográfia

1. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods, Vol. 39 (2007) 510–526
2. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics, Vol. 16 (1990) 22–29
3. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society for Information Science (JASIS), Vol. 41 (1990) 391–407
4. Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of the 4th international conference on Language Resources and Evaluation (LREC2004) (2004)
5. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. Psychological Review, Vol. 104 (1997) 211–240
6. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical cooccurrence. Behavior Research Methods, Instruments & Computers, Vol. 28 (1996) 203–208
7. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM, Vol. 18, No. 11 (1975) 613–620
8. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research (JAIR), Vol. 37 (2010) 141–188

Magyar nyelvű néprajzi keresőrendszer

Zsibrita János[1], Vincze Veronika[2]

[1] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
zsibrita@inf.u-szeged.hu
[2] MTA-SZTE Mesterséges Intelligencia Kutatócsoport
vinczev@inf.u-szeged.hu

Kivonat: A cikkben bemutatjuk Java-alapú keresőrendszerünket, mely különféle néprajzi szövegekben – hiedelmekben, táltosszövegekben és népmesékben – egész mondatos kereséseket tesz lehetővé. A rendszer azokat a dokumentumokat adja vissza, ahol a keresett ige és annak vonzatai a keresőkifejezésben megadott nyelvtani viszonyban állnak egymással. A rendszer alapjait a magyarlanc morfológiai és szintaktikai elemző moduljai jelentik. A kereső a teljes egyezések mellett részlegesen egyező találatokat is képes visszaadni, illetve a találatok grafikus megjelenítésére is van mód.

1 Bevezetés

A MASZEKER projekt keretében az angol nyelvi szabadalmi keresőrendszer mellett [1] egy magyar nyelvű néprajzi szövegeken működő keresőrendszer is elkészült. A kereső célja, hogy különféle néprajzi dokumentumokban egész mondatos kereséseket hajtson végre, azaz olyan dokumentumokat ad vissza, ahol a keresett ige és vonzatai a keresőkifejezésben megadott viszonyban állnak egymással. A keresőrendszer teljes egészében Javában implementált, így platformfüggetlenül használható.

2 A keresőrendszer

A keresőrendszer alapjait a magyarlanc morfológiai és szintaktikai (dependencia)elemző [3, 4] jelenti, amely meghatározza a szövegben levő szavak szófaját és a köztük levő nyelvtani kapcsolatokat. A keresés hátteréül szolgáló adatbázis magyar nyelvű hiedelmeket, táltosszövegeket, illetve meséket tartalmaz, összesen kb. 1,4 millió szövegszóból áll [2]. A néprajzi szövegek hatékony nyelvi elemzéséhez szükségesnek bizonyult a népies, illetve régies helyesírású szavak mai helyesírás szerinti átírása, így első lépésben ezek cseréje történt meg egy, a magyarlancba integrált szótáralapú hibajavító modul segítségével.

A keresés során a keresőmondatot először dependenciaelemzésnek vetjük alá, majd a megtalált grammatikai relációknak megfelelő illeszkedéseket keresünk a szövegekben: a keresőkifejezés igéjének összes előfordulását megkeressük, majd megnézzük, hogy az adott mondatokban levő igei bővítmények lemmája egyezik-e a

keresőkifejezésben szereplő bővítmények lemmájával, illetve hogy ugyanolyan grammatikai reláció van-e köztük (pl. mindkét esetben az ige tárgyáról van-e szó). Amennyiben igen, teljes értékű találatként adja vissza a rendszer az adott mondatot, illetve dokumentumot. Ha csak részleges egyezést tapasztalunk, például az ige egyik bővítménye egyezik, de a másik eltér, akkor azt részleges találatként jeleníti meg a rendszer. Lehetőség nyílik arra is, hogy csak az ige egyezését vizsgáljuk. A keresés során választható, mely részkorpusz szövegeiben kívánunk keresni, illetve a találatok szintaktikai reprezentációjának grafikus megjelenítésére is van lehetőség.

2.1 A keresés során használt korpusz jellemzői

A demonstráció egy magyar néprajzi korpuszon történik, aminek a konszolidálása, azaz a benne szereplő népies írásképű szavaknak a ma szokásos alakra alakítása (az eredeti íráskép megtartása mellett) már korábban megtörtént. A hiedelem- és a táltosszövegek a Néprajzi Múzeumnak a történelmi Magyarországról származó gyűjtéséből származnak, korabeli feljegyzések alapján gépelték be a kutatók. A mese korpusz néhány ingyenes internetes forrásról történő gyűjtés eredménye.

- Állatmesék (124 dokumentum)
- Formulamesék (20 dokumentum)
- Hazugságmesék (12 dokumentum)
- Legendamesék (55 dokumentum)
- Novellamesék (136 dokumentum)
- Rászedett ördög mesék (5 dokumentum)
- Rátótiádák (1 dokumentum)
- Tréfás mesék (11 dokumentum)
- Trufák és anekdoták (23 dokumentum)
- Tündérmesék (124 dokumentum)

A korpusz fontosabb adatai az 1. táblázatban láthatók.

1. táblázat: A néprajzi korpusz mérete.

Szövegtípus	Szövegek száma	Szavak száma
Népi hiedelem	2704	65807
Táltosszövegek	432	44021
Népmesék	505	633047
Összesen	**3641**	**742875**

2.2 A keresőkifejezés kialakítása

A keresés megszorított nyelvezetű keresőkifejezések alapján történik. A keresőkifejezés egy **egy tagmondatból** álló, **egy igét** tartalmazó (egyszerű) mondat. Az ige bővítményeként különféle **főnevek** szerepelhetnek különféle esetragokkal. A

határozószavak használata nem megengedett. **Tagadást** és **modalitást** jelző elemek használatát sem engedjük meg.

Néhány jólformált keresőkifejezés:

- *Foggal születik a táltos.*
- *A róka kergeti a nyulat.*
- *A lány körbefutja a házat.*

Néhány rosszulformált keresőkifejezés:

- *A vörös róka kergette tegnap a nyulat.*
- *A lány hirtelen előveszi és megeszi az almát.*
- *A róka nem eszi meg a nyulat.*
- *A róka megeheti a nyulat.*

Alárendelő mellékmondatok, illetve **mellérendelő tagmondatok** használata sem megengedett, ilyenkor több egymás utáni keresőmondatot kell alkalmazni.

- *A lány körbefutja a házat, és megeszi az almát. -> A lány körbefutja a házat. A lány megeszi az almát.*

Természetesen az alany pontos meghatározása nélkül is lehetséges keresni, ilyenkor csak az ige egyéb vonzatait tüntetjük fel a keresőkifejezésben:

- *Foggal születik.*
- *Bikával küzd.*

2.3 A keresőrendszer korlátai

A magyar nyelv grammatikai sajátosságaiból adódóan azonban problémát jelentenek a névmási referenciák, illetőleg az olyan mondatok, ahol a bővítményeket nem fejezzük ki külön szóval. Lásd az alábbi szövegrészletet:

> *Volt egyszer egy király$_i$, aki$_i$ olyan gyönyörű templomot építtetett, hogy közel s távolban nem találta senki párját. Egy szép napon azután messze távolból jött egy vándor$_j$, és (ő$_j$) hosszan bámulta a csodaszép épületet. A király$_i$ odament hozzá$_j$, és (ő$_i$) megkérdezte tőle$_j$, hogy tetszik neki$_j$ a templom.*

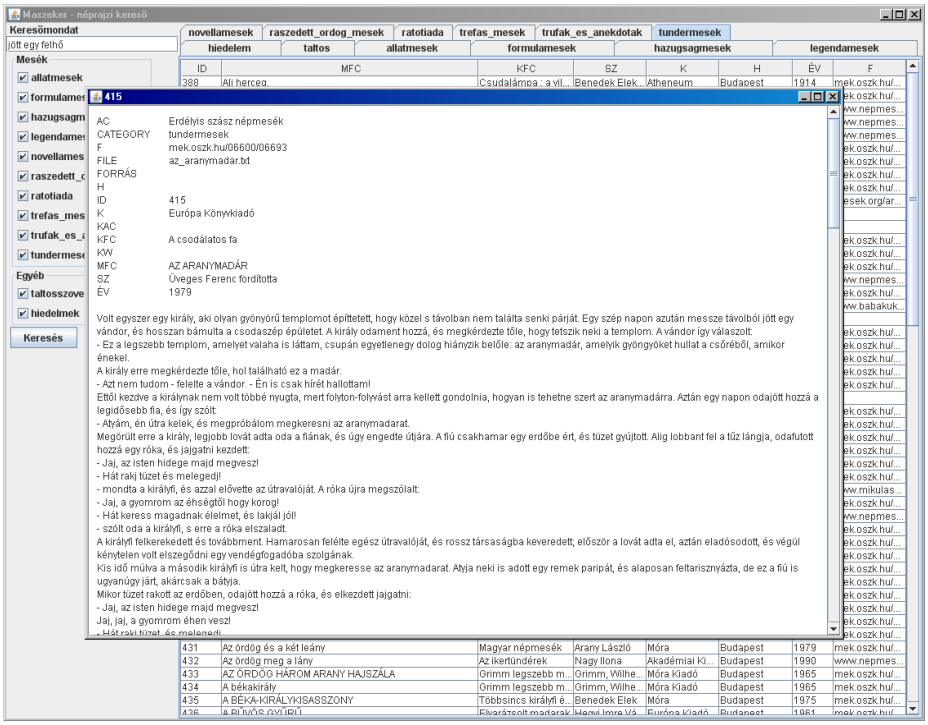
A szövegben azonos indexszel vannak jelölve az azonos egyedre utaló szavak, a zárójelbe tett névmások pedig az eredeti szövegben nem szerepelnek. Azonban legjobb tudomásunk szerint a magyar nyelvre nem áll rendelkezésre olyan automatikus elemző, amely az ehhez hasonló eseteket automatikusan azonosítaná, így jelenleg ez a

szöveg nem jelenik meg találatként az „a király odamegy a vándorhoz” keresőkifejezésre.

3 Az eredmény bemutatása

A keresés gomb megnyomásával elindul a keresés és az illesztés algoritmusa. A keresési algoritmus lefutása után megjelenik a keresőkifejezés elemzett fastruktúrája. Ezek után pedig a keresésnek megfelelő dokumentumokból készített találati lista.

Ha bármely találati listában szereplő dokumentum teljes tartalmát meg szeretnénk tekinteni, elég a dokumentum sorára kattintanunk. Ekkor egy új ablak nyílik meg, a teljes dokumentummal.



1. ábra. A találati lista egy megnyitott dokumentummal.

A megjelenő új ablakban a dokumentumra jellemző egyéb metainformációk is megjelennek, azaz annak kategóriája, az internetes forrás elérhetősége, a fájl neve, azonosítója, a kötet címe, kiadási helye és éve, a gyűjtő vagy a kötetszerkesztő neve:

AC
CATEGORY tundermesek
F
www.nepmese.hu/index.php?option=com_mtree&Itemid=26
FILE bruncik_kiralyfi.txt
FORRÁS
H Budapest
ID 459
K Akadémiai Kiadó
KAC
KFC Rózsafiú és Tulipánleány
KW
MFC Bruncik királyfi
SZ Kovács Ágnes szerk.
ÉV 1987

Az elemzett keresőkifejezés megjelenítésére is van lehetőség, l.2. ábra.

2. ábra. Egy elemzett keresőkifejezés megjelenítve.

A keresés eredménye egy új ablakban jelenik meg, dokumentumcsoportok szerint rendezve, ahogyan az a 3. ábrán is látszik. A találati lista minden sora egy-egy dokumentumot reprezentál. Minden sor tartalmazza az adott dokumentum keresőkifejezésre illeszkedő mondatát, vagyis a releváns szövegrészt.

3. ábra. A „jött egy felhő” keresőkifejezésre illeszkedő dokumentumok találati listája.

A találati lista egy tetszőleges elemére kattintva megjeleníthető a releváns mondat elemzése (l. 4. ábra). Így ellenőrizhető, hogy az algoritmus mi alapján végezte el az illesztést.

4. ábra. A találati lista egy elemének szintaktikai elemzése.

A néprajzi keresőrendszer egy népmesékben való keresést lehetővé tevő verziója kutatási célokra nyilvánosan elérhető a http://maszeker.huminf.u-szeged.hu honlapon.

Köszönetnyilvánítás

A kutatás – részben – a MASZEKER kódnevű projekt keretében a Nemzeti Fejlesztési Ügynökség támogatásával, illetve a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt keretében az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett valósult meg.

Hivatkozások

1. Szőts M., Csirik J., Gergely T., Karvalics L.: MASZEKER: projekt szemantikus keresőtechnológia kidolgozására. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Tudományegyetem, Szeged (2010) 159–167
2. Szőts, M., Darányi, S., Alexin, Z., Vincze, V., Almási, A.: Semantic processing of a Hungarian ethnographic corpus. In: Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts. Bécs, Ausztria (2010) 112–115
3. Zsibrita J., Vincze V., Farkas R.: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 275–283
4. Zsibrita J., Vincze V., Farkas R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 368–374

magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés

Zsibrita János[1], Vincze Veronika[2], Farkas Richárd[1]

[1] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
{zsibrita, rfarkas}@inf.u-szeged.hu
[2] MTA-SZTE Mesterséges Intelligencia Kutatócsoport
vinczev@inf.u-szeged.hu

Kivonat: Ebben a cikkben bemutatjuk a magyarlanc nyelvi elemző újabb változatát, amely a hatékonyabb implementációnak köszönhetően a korábban publikált verzióhoz képest jóval gyorsabban képes magyar szövegek mondatra és szövegszavakra bontására, a szavak morfológiai elemzésére, majd szófaji egyértelműsítésére a pontosság javulása mellett. A magyarlanc 2.0 továbbá tartalmaz a mondatok függőségi nyelvtan szerinti szintaktikai elemzéséért felelős modult is. A rendszer teljes egésze JAVA-ban implementált, így platformfüggetlenül használható. Az elemző kutatási célokra bárki számára szabadon hozzáférhető.

1 Bevezetés

A természetesnyelv-feldolgozás magasabb szintű alkalmazásainak elengedhetetlen alapfeltétele a szövegek mondatokra és szavakra szegmentálása, a szövegszavak morfológiai elemzése és szófaji egyértelműsítése, illetve a mondatok szintaktikai elemzése. Cikkünkben bemutatjuk a magyarlanc elemző újabb változatát, amely a hatékonyabb implementációnak köszönhetően az eddiginél jóval gyorsabban képes magyar szövegek mondatra és szövegszavakra bontására, a szavak morfológiai elemzésére, majd szófaji egyértelműsítésére, mindemellett újdonságot jelent a mondatok függőségi nyelvtan szerinti szintaktikai elemzése. A rendszer teljes egészében JAVA-ban implementált, így platformfüggetlenül használható.

2 Az elemző modulok

A korábban bemutatott magyarlanc 1.0 [11] magyar szövegek mondatra és szövegszavakra bontására, a szavak morfológiai elemzésére, majd szófaji egyértelműsítésére volt képes. Az újabb munkálatoknak köszönhetően az elemzés felgyorsult, illetőleg pontosabbá vált, továbbá a lánc kibővült a mondatok függőségi nyelvtan szerinti elemzésével, így tudomásunk szerint a magyarlanc 2.0 az első olyan eszköz, amely a szegmentálástól kezdve egészen a szintaktikai elemzésig képes végrehajtani a magyar nyelvű szövegek nyelvi előfeldolgozását.

2.1 Morfológiai elemzés

A szófaji elemző és egyértelműsítő (lemmatizáló és POS-tagger) a Stanford POS-tagger egy módosított változata, amely az ismeretlen szavakra a morphdb.hu-alapú [9] morfológiai elemző által adott lehetséges elemzéseket használja fel. Az elemzőt a kézi morfoszintaktikai annotációval rendelkező Szeged Korpuszon [2] tanítottuk, az eredeti MSD-kódok egy redukált kódhalmazán, azonban az elemzés eredményeképpen teljes értékű MSD-kódokat kapunk vissza. A kódhalmaz redukálásánál azt az irányelvet követtük, hogy a csökkentett kódkészletet használó szófaji egyértelműsítő modul kimenete egyértelműen megfeleltethető legyen az eredeti MSD-kódoknak. Tehát például az Nc-sd és Nc-sg kódok redukált alakja különbözik, míg a Nc-sd és Nc-sd---s3 ugyanarra a kódra redukálódik, mert soha nem fordulhat elő, hogy egy szóalaknak ez a két kód lehetséges elemzése (és a szófaji egyértelműsítőnek döntenie kell köztük).

A névszók (főnév, melléknév, számnév, névmás) és a nyílt tokenosztályba tartozó elemek esetében alapesetben az MSD-kódok a fő szófajra (az MSD-kód első pozíciójában álló elemre) redukálódnak. Birtokos és részes határozós esetük azonban egybeesik, ezért birtokos és részes esetben a redukált kódok megtartják az eset értékét (pl. Nd, Ng). Az essivusi *(-an/-en*, *-ul/ül*) és superessivusi (*-n/-on/-en/-ön*) esetragok szintén egybeeshetnek, pl.: *szépen*. Ezért superessivusi esetben a redukált kódok megtartják az eset attribútum értékét (pl. Ap). A névszók E/3. birtokos alakja egybeeshet a névszó birtokos nélküli alakjával, pl.: *Ajkán*. Ezért ezekben az esetekben szintén különbözőek lesznek a redukált kódok. Egy magas hangrendű névszó E/3. birtokos alakjának ragozott változata egybeeshet a névszó *-é* birtokjeles ragozott változatával, pl.: *énekét* (Nz és Ns). A névmások esetében a fenti megkötések mellett a három legfontosabb névmáscsoport (személyes, kérdő és vonatkozó) megtartja típusát is (Pe/Pq/Pr). Törtszámok esetén a redukált kódok megtartják a típust (Mf).

Alapesetben az igei MSD-kódok egyszerűen V-re redukálódnak. A segédigék kódja Va-ra redukálódik. A feltételes módú, T/1. és T/2. igék alanyi és tárgyas ragozású alakja egybeesik, pl.: *olvasnánk*. Ezért az ilyen igék tárgyas ragozású alakjainak MSD-kódja Vcp-re redukálódik. Az ikes igék E/1. alakjainak alanyi és tárgyas ragozása egybeesik, pl.: *iszom*, ezért a tárgyas ragozású MSD-kódok Vip-re redukálódnak. Feltételes módban, jelen időben, a magas hangrendű igék E/1. alanyi ragozású és T/3. tárgyas ragozású alakja egybeesik pl.: *ennék*. Ezért a T/3. tárgyas MSD-kódok alapesetben V3p-re redukálódnak. A kijelentő módú, múlt idejű, E/1. igék alanyi és tárgyas alakja egybeesik pl.: *osztottam*, ezért a tárgyas alakok MSD-kódja Vy-ra redukálódik. A felszólító módú igék kódja Vm-re redukálódik. Bizonyos esetekben egy adott ige múlt ideje és egy másik ige jelen idejű alakja egybeeshet (pl.: *ért*), ezért a korábbi szabályokra nem illeszkedő jelen idejű igék kódja Vp-re redukálódik.

Alapesetben a határozószói MSD-kódok egyszerűen R-re redukálódnak. A négy legfontosabb határozószó csoport (igekötő, kérdő, vonatkozó és személyes névmási) megtartják típusukat (Rp/Rq/Rr/Rl). A névelők MSD-kódja T-re redukálódik. A kötőszavak, névutók, indulatszavak, helyesírási hibát tartalmazó szavak, ismeretlen szavak és rövidítések esetében az eredeti MSD-kód megegyezik a redukált kóddal.

Az MSD-kódrendszer valamennyi attribútumához hozzárendeltünk egy-egy morfológiai jegyet, a magyar nyelv sajátosságainak megfelelő, a *CoNLL-2009 Shared Task*[1] kiírásnak eleget tevő módon. A szintaktikai elemző ezeket a jegyeket használja az elemzés során. Részletesen: típus – SubPOS, szám – Num, eset – Cas, birtokos száma – NumP, birtokos személye – PerP, birtok(olt) száma – NumPd, mód/forma – Mood, Idő – Tense, személy – Per, határozottság – Def, fok – Deg, klitikum – Clitic, forma – Form, mellérendelés típusa – Coord, altípus – Type. Az 1. táblázat mutatja, mely szófaj esetén mik a releváns jegyek.

1. táblázat: A szófajok és a morfológiai jegyek kapcsolata.

jegy	N	V	V	A	P	T	R	R	S	C	M	I	I	X	Y	Z	O	O
Sub-POS	•	•	•	•	•	•	•	l	•	•	•		o				•	e|d|n
Num	•	•	•	•	•			•			•						•	•
Cas	•			•	•						•						•	•
NumP	•			•	•						•						•	•
PerP	•			•	•						•						•	•
NumPd	•			•	•						•						•	•
Mood		•	n															
Tense		•																
Per		•	•		•			•										
Def		•																
Deg				•			•	•										
Clitic																		
Form										•	•							
Coord										•								
Type																		•

2.2 Szintaktikai elemzés

A szintaktikai elemzésnek két a leggyakoribb reprezentációs módja a konstituensfa és a függőségi fa. A függőségi fákkal dolgozó elemzők különösen jól használhatóak szabad szórendű nyelvek elemzésére, így a magyarra is, ezek ugyanis könnyebben teszik lehetővé az egymással nem szomszédos, de összetartozó szavak összekapcsolását is.

A függőségi elemzők két fő megközelítésre épülnek. A gráfalapú modellek a mondat szavait mint csúcspontokat tartalmazó teljes gráfon keresik a legvalószínűbb feszítőfát [3,7]. A tranzakcióalapú modellek balról jobbra haladva szavanként elemzik a mondatot [6,8]. A magyarlancba beépítendő függőségi elemző kiválasztása előtt három függőségielemző-implementációval is kísérleteket végeztünk: egy átmenetalapú modellt (Malt [8]) és két gráfalapú modellt (MST [7] és Bohnet-parser [1]) vizsgáltunk [4].

A mérések alapjául a Szeged Dependencia Treebank [10] szolgált. A treebank eredeti változatában a többtagú tulajdonnevek össze voltak vonva, azaz egy tokenként

[1] http://ufal.mff.cuni.cz/conll2009-st/task-description.html

voltak kezelve. A valóságban azonban nem létezik olyan algoritmus, amely hiba nélkül vonja össze a többtagú tokeneket, így méréseinkhez mi is több részre bontottuk ezeket. Az új tokenek tulajdonnévi kódot kaptak, alapértelmezett morfológiai jegyekkel, kivéve az utolsó tokent, amely megtartotta az eredeti elemzést. Így például a *Kovács és társa kft.* frázis az új annotációban N N N N szófaji kódokat kapott (megjegyezzük, hogy a Penn Treebank annotációs elveit követve N C N N kódokat kellett volna kapnia, azaz a tulajdonnévben előforduló kötőszó kötőszói kódot kap). A tulajdonnevek belső szintaktikai szerkezetét nem jelöltük be: egy láncot alkotnak az első tagtól az utolsóig. E döntések hátterében az áll, hogy legjobb tudomásunk szerint nem léteznek olyan alkalmazások, amelyek hasznosítani tudják a tulajdonnevek belső szerkezetét.

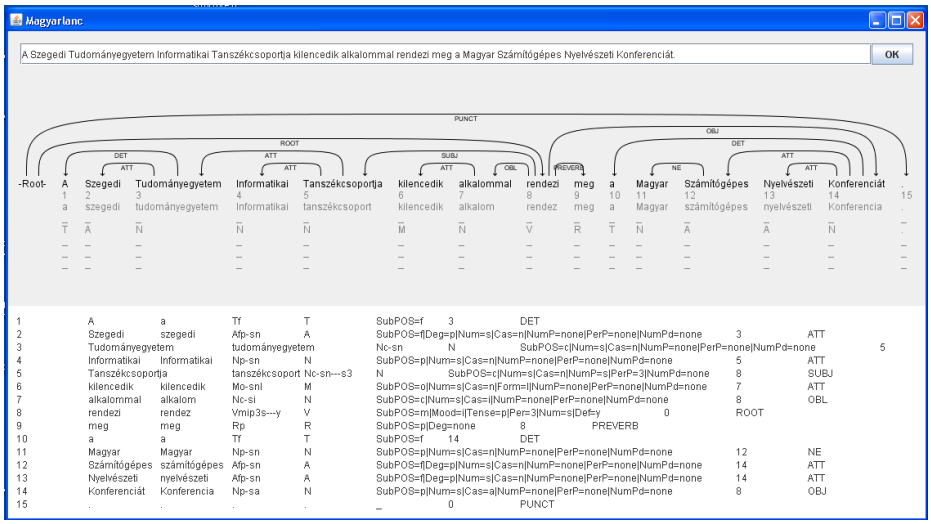
Az ige nélküli tagmondatok (túlnyomórészt névszói állítmány) esetében a Szeged Dependencia Treebank virtuális csomópontokat alkalmaz (16 000 előfordulás). A megoldás előnyei közé tartozik, hogy így hasonló faszerkezetet tulajdonítunk a mondatnak kijelentő mód jelen idő egyes/többes szám harmadik személyben, mint más módban, időben és számban/személyben.. A jelenleg elérhető szintaktikai elemzők azonban nem képesek a virtuális csomópontok megfelelő kezelésére. Éppen ezért, összhangban a Prague Dependency Treebankben alkalmazott megoldással [5], a virtuális csomópontokat töröltük a fából, és gyermekeiket a virtuális csomópont szülő csomópontjához kötöttük, illetve az Exd címkével láttuk el. Amennyiben a mondat gyökéreleme virtuális csomópont volt, ennek törlése azt eredményezte, hogy a mondatban nem maradt gyökérelem, aminek következtében az ilyen mondatokat kiszűrtük a korpuszból, és kísérleteinkben nem használtuk fel őket.

Mivel a kísérletek eredményei alapján a Bohnet-parser bizonyult a legpontosabbnak és leghatékonyabbnak, így ezt a függőségi elemzőt integráltuk az elemző láncba. A magyarlanc 2.0-ba integráltuk a whatswrong[2] megjelenítőt is, így a szintaktikailag elemzett mondatok ágrajzának vizuális megjelenítésére is van lehetőség.

2.3 Az elemző lánc kimenete

Az 1. ábra bemutatja az elemző felhasználói felületét egy mintaelemzéssel. A képernyő közepén látható a mondat függőségi elemzése, majd a képernyő alján található a részletes morfológiai elemzés.

[2] https://code.google.com/p/whatswrong/



1. ábra. A magyarlanc 2.0 felhasználói felülete.

Az elemzett kimeneti fájlok felépítése a következő. Egy sor egy tokennek felel meg, a mondatokat üres sorok választják el egymástól. Az első oszlopban a token mondatbeli sorszáma, a másodikban a szóalak, a harmadikban a lemma, a negyedikben az MSD-kód, az ötödikben a szófaj, a hatodikban a morfológiai jegyek, a hetedikben a szülő csomópont sorszáma, a nyolcadikban pedig a függőségi élcímke látható. Az alábbiakban közlünk egy példát a kimeneti fájlformátumra.

```
1    Az      az      Tf      T       SubPOS=f        2       DET
2    elnök   elnök   Nn-sn   N
     SubPOS=n|Num=s|Cas=n|NumP=none|PerP=none|NumPd=none 3       SUBJ
3    megígérte       megígér Vmis3s---y      V
     SubPOS=m|Mood=i|Tense=s|Per=3|Num=s|Def=y    0       ROOT
4    ,       ,       ,       ,       _       3       PUNCT
5    az      az      Tf      T       SubPOS=f        7       DET
6    észlelt észlelt Afp-sn  A
     SubPOS=f|Deg=p|Num=s|Cas=n|NumP=none|PerP=none|NumPd=none  7
     ATT
7    hibákat hiba    Nn-pa   N
     SubPOS=n|Num=p|Cas=a|NumP=none|PerP=none|NumPd=none 14      OBJ
8    a       a       Tf      T       SubPOS=f        9       DET
9    szövetség       szövetség       Nn-sn   N
     SubPOS=n|Num=s|Cas=n|NumP=none|PerP=none|NumPd=none 10      ATT
10   vezetése        vezetés Nn-sn---s3      N
     SubPOS=n|Num=s|Cas=n|NumP=s|PerP=3|NumPd=none       14      SUBJ
11   45      45      Mc-snd  M
     SubPOS=c|Num=s|Cas=n|Form=d|NumP=none|PerP=none|NumPd=none 12
     ATT
12   napon   nap     Nn-sp   N
     SubPOS=n|Num=s|Cas=p|NumP=none|PerP=none|NumPd=none 13      OBL
13   belül   belül   St      S       SubPOS=t        14      TLOCY
14   kijavítja       kijavít Vmip3s---y      V
     SubPOS=m|Mood=i|Tense=p|Per=3|Num=s|Def=y    3       ATT
15   .       .       .       .       _       0       PUNCT
```

3 Eredmények

A magyarlanc 2.0 elemzési pontosságát megállapítandó kísérleteket végeztünk mind a szófaji egyértelműsítés, mind a szintaktikai elemzés terén. Méréseinkhez a Szeged Dependencia Treebanket [10] használtuk fel. A treebank mondatait véletlenszerűen osztottuk fel tanító (80%) és kiértékelési (20%) adatbázisra. Az alábbiakban ezen mérések eredményeit ismertetjük.

3.1 A szófaji egyértelműsítés eredményei

A szófaji egyértelműsítés a tesztadatbázison 96,33%-os pontosságot ért el. Az átalakításoknak köszönhetően a korábbi magyarlanc 1.0 verzióhoz képest pontosabbá vált a számok és a nyílt tokenosztályba tartozó tokenek elemzése.

3.2 A szintaktikai elemzés eredményei

A szintaktikai elemzés kiértékeléséhez a Labeled Attachment Score (LAS) és az Unlabeled Attachment Score (ULA) metrikákat használjuk. A LAS esetében a teljes egyezéshez szükséges mind a szülő, mind az élcímke egyezése az etalonhoz képest, míg az ULA esetében elégséges a szülő csomópont egyezése (itt nem számít hibának a rossz élcímke). A függőségi elemzés a tesztkorpuszon 91,42%-os (LAS) és 93,22%-os (ULA) eredményt ért el.

3.3 Az elemzés sebessége

A magyarlanc jelen verziójának működési sebességét az Egri csillagok című regényen teszteltük. A teljes elemzési láncot futtatva 1 GB RAM felhasználásával percenként 1000 mondat elemzése történik meg. Amennyiben csak szófaji egyértelműsítést szeretnénk végezni, az 3000 mondat/perc sebességgel zajlik, ami a korábban publikált 1.0 verzióhoz képest harmincszoros gyorsulást jelent.

4 Összegzés

Cikkünkben bemutattuk a magyarlanc 2.0 elemző láncot, amely magyar nyelvű szövegek nyelvi előfeldolgozására – szegmentálás, morfológiai elemzés, szófaji egyértelműsítés és szintaktikai (függőségi nyelvtan szerinti) elemzésre – hivatott, és a korábban publikált verzióhoz képest jóval gyorsabban képes minderre megnövekedett pontosság mellett. A rendszer teljes egészében JAVA-ban implementált, így platformfüggetlenül használható. Az elemző lánc kutatási célokra szabadon hozzáférhető a http://www.inf.u-szeged.hu/rgai/magyarlanc oldalon.

Köszönetnyilvánítás

A kutatás a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt keretében az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett valósult meg.

Hivatkozások

1. Bohnet, B.: Top accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) (2010) 89–97
2. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the Eighth International Conference on Text, Speech and Dialogue (TSD 2005). Karlovy Vary, Czech Republic 12-16 September, and LNAI series Vol. 3658 (2005) 123-131
3. Eisner, J. M.: Three new probabilistic models for dependency parsing: an exploration. In: Proceedings of the 16th Conference on Computational Linguistics - Volume 1, COLING '96 (1996) 340–345
4. Farkas, R., Vincze, V., Schmid, H.: Dependency Parsing of Hungarian: Baseline Results and Challenges. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (2012) 55-65
5. Hajič, J., Böhmová, A., Hajičová, E., Vidová-Hladká, B.: The Prague Dependency Treebank: A Three-Level Annotation Scenario. In: Abeillé, A. (ed.): Treebanks: Building and Using Parsed Corpora. Amsterdam, Kluwer (2000) 103–127
6. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: Proceedings of the 6th Conference on Natural Language Learning - Volume 20, COLING-02 (2002) 1–7
7. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005) 523–530
8. Nivre, J., Hall, J., Nilsson, J.: Memory-Based Dependency Parsing. In: HLTNAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004) (2004) 49–56
9. Trón, V., Halácsy, P., Rebrus, P., Rung, A., Vajda, P., Simon, E.: Morphdb.hu: Hungarian lexical database and morphological grammar. In: Proceedings of 5th International Conference on Language Resources and Evaluation (2006)
10. Vincze, V. Szauter, D., Almási, A., Móra, Gy., Alexin, Z., Csirik, J.: Hungarian Dependency Treebank. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (2010)
11. Zsibrita J., Vincze V., Farkas R.: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged, Szegedi Tudományegyetem (2010) 275-283

Szerzői index, névmutató