Sulyok Hedvig Juhász Valéria Erdei Tamás

Beszéd- és nyelvelemző szoftverek

a versenyképességért és az esélyegyenlőségért

HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban

SZTE JGYPK
Szeged, 2019

Emberi Erőforrások Minisztériuma
Emberi Erőforrás Támogatáskezelő
Nemzeti Együttműködési Alap

Sulyok Hedvig Juhász Valéria Erdei Tamás

Beszéd- és nyelvelemző szoftverek

a versenyképességért és az esélyegyenlőségért

HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban

SZTE JGYPK
Magyar és Alkalmazott Nyelvészeti Tanszék
Szeged, 2019

Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért
HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban

Az SZTE JGYPK-n 2018. október 19-én rendezett konferencia tanulmánykötete

Szerkesztők:
SULYOK HEDVIG, JUHÁSZ VALÉRIA, ERDEI TAMÁS

Olvasószerkesztő:
SULYOK HEDVIG

Technikai szerkesztő:
ERDEI TAMÁS

Borítóterv:
HERASZKA VIKTÓRIA

Közreműködő szervezetek:

Szegedi Tudományegyetem Juhász Gyula Pedagógusképző Kar
Magyar és Alkalmazott Nyelvészeti Tanszék

Magyar Alkalmazott Nyelvészek és Nyelvtanárok Egyesülete

Emberi Erőforrások Minisztériuma
Emberi Erőforrás Támogatáskezelő
Nemzeti Együttműködési Alap

CLARIN – HunCLARIN

A rendezvény és a kötet létrejöttét a Nemzeti Együttműködési Alap (pályázati azonosító: NEA-KK-18-SZ-0653) és a CLARIN ERIC támogatta.

ISBN: 978-615-5455-93-3

Kiadja:
SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék
Szeged, 2019

Tartalomjegyzék

Előszó

Ennek a kis kötetnek a címe két részből áll. Az első rész a tulajdonképpeni tartalom: *beszéd- és nyelvelemző szoftverek.* A második rész, a folytatás már szokatlanabb: *a versenyképességért és az esélyegyenlőségért*. Ráadásul még van egy alcím is: *HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Ezeknek a címrészleteknek az egyes részei magyarázatra szorulnak.

A beszéd- és nyelvelemző szoftverek pontosan azok a számítógépeken futó megoldások, melyek az emberi beszéd vagy az ember által előállított szöveg nyelvészeti elemzését végzik abból a célból, hogy további szoftverek az így feldolgozott nyelvi információ segítségével a hangzó vagy leírt szövegekben kódolt információt minél hatékonyabban tovább tudják adni a számítógéppel dolgozó nemnyelvész felhasználók számára.

A cím második részében megjelenő két szó, a *versenyképesség* és az *esélyegyenlőség* azt jelzi, hogy az efféle számítógépes nyelvészeti megoldások lehetővé teszik a napi feladatok hatékonyabb elvégzéséből következő versenyképesség-növekedést, ráadásul az ezeket az eszközöket használó, nyelvi, hallási vagy egyéb értelmezési nehézségekkel küzdő felhasználók kevésbé kerülnek hátrányba a beszéd- vagy szövegértelmezés során az ezekkel a problémákkal nem rendelkező embertársaikhoz viszonyítva.

Végül az alcímben szereplő *HunCLARIN* egy szervezet, a CLARIN nemzetközi számítógépes nyelvészeti hálózat magyar tagja, mely a hazai nyelvtechnológia minden fontos szereplőjét tömöríti. Ami különösen hatékonnyá teszi a működését, az az, hogy a tagintézményeiről, melyek kutatócsoportok és vállalkozások – nyelvészeti szakkifejezéssel – azt lehet elmondani, hogy komplementer disztribúcióban vannak, azaz alig van egymást átfedő kutatási terület. Ennek az egymást jól kiegészítő közösségnek a fóruma volt az az esemény is, melynek írásos anyagait adjuk közre ebben a kötetben.

Prószéky Gábor

Bevezetés a korpuszok és nyelvi adatbázisok világába

Vincze Veronika
tudományos főmunkatárs
MTA-SZTE Mesterséges Intelligencia Kutatócsoport
vinczev@inf.u-szeged.hu

Elméleti nyelvészetből és informatikatudományból doktoráltam a Szegedi Tudományegyetemen. Jelenleg számítógépes nyelvészként dolgozom az MTA-SZTE Mesterséges Intelligencia Kutatócsoportban, feladatom elsősorban a csoport projektjeinek nyelvészeti felügyelete és koordinálása. Érdeklődési körömbe elsődlegesen a korpuszépítés és a többszavas kifejezések számítógépes kezelése tartozik, de foglalkozom számítógépes morfológiával és szintaxissal, emellett információkinyeréssel is.

1. Bevezetés

A nyelvészeti kutatásokban jó ideje megkülönböztetik a kompetencia és performancia fogalmát (Chomsky 1957). Egy nyelv anyanyelvi beszélői kompetenciájuk segítségével képesek jól formált mondatokat alkotni az adott nyelven, így tudják eldönteni, hogy egy adott nyelvi megnyilatkozás megfelel-e a nyelv szabályainak vagy sem. A performancia ezzel szemben a nyelv gyakorlati megvalósulását jelenti: amit egy beszélő egy adott pillanatban kimond. Bizonyos esetekben a performancia nem követi a kompetenciát: ha például a beszélő fáradt vagy dekoncentrált, akkor elkövethet nyelvbotlásokat, megnyilatkozásában nem mindig követi a nyelv adott szabályait.

A nyelvészeti kutatások általában kétféle módszertannal dolgoznak: vannak adatorientált és elméletorientált módszerek. Az elméletorientált módszerek elsődlegesen a kompetenciára épülnek, azaz azt vizsgálják, az adott nyelvben mi lehetséges és mi nem, milyen szerkezetek lehetségesek és mik nem a nyelvi kompetenciának megfelelően. Vizsgálati módszereik igen gyakran épülnek introspekcióra, azaz a kutató a saját nyelvérzékére (intuíciójára) építve alkot lehetséges példamondatokat, melyeket aztán más anyanyelvi beszélőkkel véleményeztet, természetességüket, elfogadhatóságukat megítélendő.

Ezzel szemben az adatorientált módszerek a már létező nyelvi adatokból indulnak ki, ezeket elemzik, csoportosítják, ezeket próbálják meg szabályokkal leírni. A nyelvi adatokat a kutatók gyűjthetik adatközlőktől, például kérdőíves fel-

mérések vagy interjúk segítségével. Ezen felül a nyelvi adatok származhatnak adatbázisokból, szöveggyűjteményekből (azaz korpuszokból) is.

A korpusz ténylegesen előforduló írott vagy lejegyzett beszélt nyelvi adatok gyűjteménye. Általában speciális célokra hozzák létre őket, és a szövegek gyakran egy adott témakör köré csoportosulnak. A szövegeket valamilyen szempont szerint válogatják és rendezik. Nem feltétlenül egész szövegek vannak benne, és nem csak tárháza a szövegeknek, hanem sok esetben úgynevezett annotációt is tartalmaz: a szövegekben akár automatikus, akár kézi úton különféle nyelvi információk vannak jelölve, emellett a szövegek bibliográfiai adatai, szerkezeti egységei is eltárolódnak. A számítógépek kapacitásának megsokszorozódása révén a nagy méretű korpuszok összeállítása, tárolása és feldolgozása már megvalósítható, sőt kívánatos. A korpuszban található nyelvi adatok elemzése a korpusznyelvészet feladata.

E tanulmány célja, hogy az olvasót megismertesse néhány korpusszal és egyéb nyelvészeti adatbázissal, továbbá a korpusznyelvészet alapjaival. A legfontosabb alapfogalmak után ismertetjük a különféle korpusztípusokat, létrehozási módjukat, továbbá néhány példán keresztül megmutatjuk, milyen nyelvészeti jellegű információkat (annotációkat) tudunk a szövegekben kódolni. Arra is hozunk példát, hogy a nyers szövegállományból miként tudunk automatikusan annotált adatbázist előállítani. A korpuszok gyakorlati felhasználására is külön figyelmet fordítunk: bemutatjuk, hogy a korpuszokból származó adatokat hogyan lehetséges kigyűjteni, majd azokat nyelvészeti vagy más bölcsészettudományi kutatásra felhasználni.

2. Korpusztípusok

A korpuszokat számos szempont alapján csoportosíthatjuk: a szövegek nyelve szerint, modalitás szerint, a szövegek műfaja szerint stb. Modalitás szerint beszélhetünk írott nyelvi korpuszokról, melyek különféle szövegeket tartalmaznak, beszédkorpuszokról, melyek hanganyagokat és ezek szöveges átiratait foglalják magukban. Manapság pedig egyre nagyobb a multimodális korpuszok jelentősége is, melyek akár videófelvételeket is tartalmazhatnak, ezáltal hangi, képi és szöveges adatok is szerepelnek bennük.

Míg a korpuszok egy része egynyelvű dokumentumokból áll, addig számos korpusz két vagy több nyelven is tartalmaz(za ugyanazokat a) dokumentumokat. A párhuzamos korpuszokban ugyanannak a szövegállománynak többnyelvű

megfelelői vannak bekezdés, mondat és/vagy kifejezés szintjén megfeleltetve egymásnak: a világ egyik legnagyobb párhuzamos korpusza például a Biblia, melyet a világ számos nyelvére fordítottak már le.

A korpuszok – a szövegek tematikáját tekintve – lehetnek homogének, illetve heterogének, szintén az adott cél függvényében. Dönthetünk úgy, hogy minél nagyobb területet szeretnénk lefedni a nyelvi spektrumból, így több forrásból és témakörből választunk ki szövegeket. Ilyen például a Magyar Nemzeti Szövegtár (lásd lejjebb), amelynek készítői a sajtó, szépirodalom, tudományos, hivatalos és személyes stílusrétegéből válogattak szövegeket, odafigyelve arra is, hogy a határon túli nyelvváltozatok is képviselve legyenek a korpuszban. Ha azonban egy speciális alkalmazáshoz készítünk korpuszt, akkor igen gyakran behatárolt a témakör, például ha betegek dohányzási szokásait szeretnénk automatikusan kinyerni a kórlapokban rejlő információk alapján, magától értetődően orvosi jellegű dokumentumokat kell beépíteni a korpuszba. A szövegek kiválasztásakor arra is ügyelnünk kell, hogy az minél reprezentatívabb legyen az adott területre, azaz a szövegek rendelkezzenek a területre jellemző nyelvi és formai sajátosságokkal.

Beszélők vagy szerzők szerint is csoportosíthatjuk az adott szövegeket. Például egy korpusz tartalmazhat tájnyelvi szövegeket, ahol egy adott tájegységben élőktől származó nyelvi produktumokat gyűjtünk össze (például erdélyi magyar adatközlőktől gyűjtött szövegek). Fókuszálhatunk a gyermeki nyelvhasználatra a gyermeknyelvi korpuszok segítségével (lásd Babarczy 2019), illetve a nyelvtanulói korpuszokat használva felderíthetjük például a magyart mint idegen nyelvet tanulók számára nehezebb, problémásabb nyelvi jelenségeket (lásd Durst et al. 2013). Kitekintve más bölcsészettudományok felé, egy adott író vagy költő összes művei is tekinthetők egy írói korpusznak, lehetőséget adva mélyebb stilisztikai vagy egyéb irodalomtudományi elemzésekre.

Összeállíthatunk egy korpuszt egy adott nyelvi stílusréteg vagy regiszter szerint is, akár szakmai nyelvhasználatra való tekintettel is. Az utóbbira egy példa a Miskolc Jogi Korpusz vagy a SZEMEK orvosi szaknyelvi korpusz (Vincze 2018). Szempont lehet a szövegek kiválasztásában a szövegek keletkezései ideje, például a nyelvtörténeti, nyelvemlékeket tartalmazó korpuszok jöttek így létre (Simon 2019).

Természetesen léteznek olyan korpuszok is, melyek heterogén adatokat tartalmaznak, azaz több szövegtípusból, stílusrétegből és műfajból, valamint több szerzőtől származó szövegek is megjelennek az anyagban. Az ilyen általános célú

korpuszok esetében gyakran az adott nyelv vagy nyelvi réteg minél teljesebb reprezentációja a cél. Ezek a korpuszok sokszor nagyobb méretűek, jellemzően több millió szövegszót tartalmaznak, mint például a Magyar Nemzeti Szövegtár vagy a Szeged Korpusz (lásd lejjebb).

Az alábbiakban a teljesség igénye nélkül felsorolunk néhány ismertebb, az angol és a magyar nyelvre vonatkozó korpuszt.

A legnagyobb méretű, angol nyelvű szövegeket tartalmazó korpuszok az alábbiak: British National Corpus (BNC), Wall Street Journal (WSJ), Reuters. Ezek körülbelül 100 millió szövegszót tartalmaznak; a dokumentumok, bekezdések határai jelölve vannak bennük, egyéb (nyelvi) annotációt azonban nem foglalnak magukban. A Gigaword korpusz körülbelül 2 milliárd szóból áll, ez sem tartalmaz nyelvi annotációt – már méreténél fogva sem. A nyelvi annotációt tartalmazó angol nyelvű korpuszok közül a legismertebb a Penn TreeBank, mely 5 millió szövegszóból áll. A szavak szófaji kódja (POS-tag) meg van adva, és szintaktikai elemzés (konstituensfa) is található a korpusz mondataihoz.

A Magyar Nemzeti Szövegtár (Oravecz et al. 2014) a mai magyar írott köznyelv általános célú reprezentatív korpusza, amely a magyarországiak mellett a határon túli magyar nyelvváltozatokat is felöleli. Jelenleg több mint egymilliárd szövegszót tartalmaz. Az MNSZ lényegi tulajdonsága, hogy minden szó mellett feltünteti a szótövet, a szófajt és a szó morfológiai elemzését is. A szótő, szófaj és elemzés megállapítása és az elemzések egyértelműsítése automatikus gépi eszközökkel történik. A korpuszban való kereséshez külön online felület áll rendelkezésre (vö. Sass 2019).

A Szeged Korpusz és Treebank a legnagyobb, kézzel egyértelműsített magyar nyelvű adatbázis, melyben 1,2 millió szövegszó található hat különböző doménből (Csendes et al. 2005). A szövegek morfológiai és szintaktikai kézi elemzéssel rendelkeznek, valamint egyes részkorpuszokon további szemantikai annotációk (pl. tulajdonnevek) is elkészültek. A részletes kézi annotálásnak köszönhetően a treebank különböző verziói megbízható tanulási és tesztelési adatbázisként szolgálnak számítógépes tanulóalgoritmusok számára.

3. Annotáció

A legtöbb korpusz nem pusztán nyers szövegekből áll: általában be vannak jelölve a szöveg szerkezeti részei is, azaz szakaszokra, bekezdésekre, mondatokra, szövegszavakra (tokenekre) van bontva. Emellett többnyire annotációt is tartal-

maznak: az annotálási munkálatok során (nyelvész) szakértők – vagy automatikus annotáció esetében egy algoritmus – kézzel bejelölik a releváns információkat a szövegállományokban, például minden egyes szóhoz hozzárendelik a szófaját vagy a szövegben megjelölik a tulajdonneveket.

Az annotáció lehet dokumentumszintű (például egy e-mail spam-e, vagy sem), mondatszintű (például a mondat tényszerű információkat közöl-e, avagy bizonytalan, esetleg tagadott információt tartalmaz), illetve szószintű (például morfológiai elemzés). Egy korpuszban természetesen többféle annotáció is szerepelhet egyidejűleg, hiszen akár többszintű (morfológiai, szintaktikai és szemantikai) nyelvi elemzést is tartalmazhat egy adott korpusz. Mindemellett vannak annotáció nélküli korpuszok is: ezeket általában statisztikai célokra, például szógyakoriság megállapítására lehet hasznosítani (hányszor fordul elő egy adott szóalak egy kellően nagy korpuszban).

Az annotáció során (nyelvész) szakértők – előre meghatározott irányelvek alapján – kézzel bejelölik a szövegekben a releváns információkat, illetve ellenőrzik a gépi annotáció minőségét és kézzel javítják annak hibáit. Az annotálás módszertanát tekintve az annotáció lehet:

– egyszeres: egy szövegen egy annotátor megy végig;
– többszörös: egyazon szövegen több annotátor is teljes egészében végigmegy, egymástól függetlenül. Amennyiben eltérés mutatkozik a két (vagy több) annotáció között, egy újabb független annotátor dönt (egyértelműsít) a problémás esetekben.

A többszörös annotáció, noha időigényesebb és drágább, általában javítja az annotáció minőségét, hiszen több szakértő nézi át ugyanazt az anyagot. Előnyei közé tartozik még, hogy lehetővé teszi az egyetértési arány mérését is: az annotátorok által egyformán jelölt esetek százalékos arányát a gépi alkalmazások által elérhető felső határnak szokták tekinteni, így voltaképpen a feladat nehézségi fokának jelzésére is alkalmas ez a mérőszám. Az egyszeres annotáció előnyeként említhető, hogy olcsóbb és gyorsabb, mint a többszörös annotáció, azonban hátránya, hogy esetenként kevésbé pontos annotációt eredményez, és nem lehetséges vele egyetértési arányt mérni.

4. Korpuszépítés

Amennyiben nyelvészeti kutatásunkhoz korpuszból kívánunk adatokat gyűjteni, felmerül a kérdés, milyen korpuszt használjunk. Első kérdésként érdemes

megvizsgálnunk, hogy az adott kutatási témához illeszkedő korpusz elérhető-e számunkra. Ha rendelkezésre áll a céljainknak megfelelő korpusz, akkor elégséges lehet a meglevő korpuszból kigyűjteni a megfelelő adatokat. A korpuszban való keresési technikákról bővebben lásd Sass Bálint e kötetbeli tanulmányát (Sass 2019), a magyar nyelvű kereshető korpuszokról pedig a Nemzeti Korpuszportálon (http://corpus.nytud.hu/nkp) találunk bővebb információt.

Ha még korábban nem hoztak létre a céljainknak megfelelő korpuszt, akkor érdemes megfontolni a saját korpusz építését. Egy korpusz megtervezésekor és létrehozásakor számos szempontot kell mérlegelni. El kell döntenünk, hogy milyen célra kívánjuk használni a korpuszt – ennek ugyanis lényegi szerepe van a szövegek kiválasztásában, a korpusz méretének meghatározásában, az annotációs elvek kidolgozásában stb. Amennyiben a korpuszt tanító- vagy tesztadatbázisként szeretnénk hasznosítani algoritmusok fejlesztéséhez, fontos a megfelelő méret: elegendő nagynak kell ahhoz lennie, hogy kellő mennyiségű példát (és ellenpéldát) szolgáltasson az adott jelenségre. Az, hogy mi számít megfelelő méretnek, mindig az adott feladat függvénye: egy tulajdonnév-felismerő rendszer betanításához általában elegendő egy néhány százezer szövegszavas annotált korpusz (például Szeged NE korpusz, Szarvas et al. 2006), azonban egy szintaktikai elemző betanítása már milliós nagyságrendű szövegszóból álló annotált korpuszt igényel (például Szeged Treebank, Csendes et al. 2005).

A szövegek gyűjtéséhez el kell döntenünk a szövegek tematikáját (például jogi vagy irodalmi szövegeket szeretnénk vizsgálni). Döntést kell hozni a kutatni kívánt nyelvi regiszterekről is (például hivatalos nyelv, köznyelv, internetes nyelvhasználat…), valamint egyéb jellemzőkről is (például a szövegek keletkezési ideje vagy szerzője szerint is szűkíthetjük a kutatott szövegek halmazát). Nem elhanyagolható szempont a szövegek hozzáférhetősége sem, azaz egyrészt magunk hozzáférünk-e könnyen a korpuszba illeszteni kívánt szövegekhez, másrészt pedig hogy milyen módon tehetjük azokat hozzáférhetővé mások számára. Itt külön felhívnánk a figyelmet a szerzői jogokra – például irodalmi szövegek esetén –, illetve bizonyos szövegtípusok, különösen az orvosi és jogi dokumentumok megkövetelik a bennük szereplő érzékeny adatok anonimizálását.

A korpuszba bekerülő szövegek összegyűjtését azok gépi előfeldolgozása, illetve – amennyiben szükséges – digitalizálása követi. Az állományok automatikus megtisztítása, szakaszokra, bekezdésekre, mondatokra és tokenekre bontása után következhet az annotálási fázis (lásd részletesebben fent). A korpuszépítés utómunkálataiként megtörténik az annotált állományok összefésülése, a formai

hibák (automatikus és/vagy kézi) javítása, majd ezek után következhet a korpusz használatbavétele.

5. A korpuszok felhasználhatósága

A korpuszokat referencia-adatbázisként különböző alkalmazások tesztelésére szokás használni: a kézi annotációt etalonnak tekintve számszerűsíteni lehet, mennyire teljesít jól az adott rendszer (kiértékelés).

A tesztelés mellett a korpuszokat az algoritmusok betanítására is lehet használni. A tanítás során a szakértő példákat mutat az algoritmusnak az annotált korpuszból, amelyek alapján az algoritmus automatikusan állítja elő a szabályokat. Az algoritmus célja, hogy a tanult szabályok használatával a korábban nem látott példányokat is megtalálja / felismerje / azonosítsa. A feladattól függően számos példára lehet szükség a hatékony tanuláshoz.

6. Adatgyűjtés programozás nélkül: készítsünk szófelhőt!

Az alábbiakban bemutatjuk, hogyan tudunk könnyen és gyorsan látványos adatvizualizációt készíteni. Ehhez nincs szükség programozási tudásra, átlagos számítógép-felhasználói ismeretek segítségével is könnyen elboldogulunk.

Szerencsére már olyan elemző eszközök is rendelkezésre állnak, melyek programozói ismeretek nélkül is képesek támogatni a korpusznyelvészet iránt érdeklődőket. Az alábbiakban bemutatunk néhány olyan eszközt, melyek szövegek nyelvi elemzését, részletesebben: mondatra és szövegszavakra bontását, azok szófaji egyértelműsítését és morfológiai, valamint szintaktikai elemzését valósítják meg. E tanulmányban a magyarlanc és az UDPipe eszközöket mutatjuk be, de a kötetben Mittelholcz Iván tanulmánya részletesen is ismerteti az e-magyar eszközt, mely hasonló funkciókkal bír (Mittelholcz 2019).

A magyarlanc nevű nyelvi előfeldolgozó eszköz a Szegedi Tudományegyetem fejlesztése (Zsibrita et al. 2013). Egy magyar nyelvű szöveges állományból kiindulva (txt) képes a szöveg mondatokra és szavakra bontására, a szavak morfológiai elemzésére, majd szófaji egyértelműsítésére, továbbá kétféle szintaktikai elemzést is képes hozzárendelni a mondatokhoz, választhatóan függőségi (dependencia) nyelvtani elemzést vagy pedig összetevős elemzést. A magyarlanc elérhető a https://rgai.inf.u-szeged.hu/node/100 oldalon, az innen letölthető program segítségével txt formátumú szövegfájlok elemzése is lehetséges parancssorból. Ha pedig csak egy-egy mondat elemzésére van szükségünk, vagy pusztán

tesztelni szeretnénk az alkalmazást, erre a http://rgai.inf.u-szeged.hu/magyarlanc-service oldalon elérhető online demó nyújt lehetőséget.

A http://lindat.mff.cuni.cz/services/udpipe/ honlapon ingyenesen elérhető UDPipe nevű elemző a Universal Dependencies annotációs sémán alapul (Straka és Straková 2017), mely egy nemzetközileg egységes morfológiai és szintaktikai annotációs séma, jelenleg kb. 50 nyelvre – köztük magyarra – dolgozták ki. A magyarlanchoz hasonlóan képes a nyers szövegek mondatra és szavakra bontására és szófaji elemzésére, továbbá a mondatok függőségi elemzésére. Egy-egy mondat és szövegfájl elemzését egyaránt lehetséges elvégezni online a fenti honlapon.

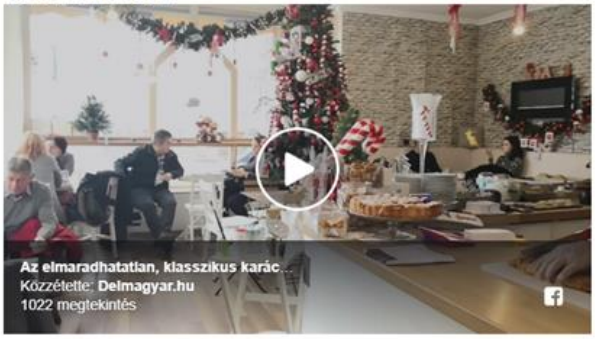
A két nyelvi elemző hasonló funkciókkal rendelkezik. Hogy a kettő közti választást elősegítsük, felsorolunk néhány további szempontot. Technikai oldalról talán könnyebben kezelhető az UDPipe, azonban kevéssé pontos elemzési eredményt ad a rendszer, mivel néhány ezer mondatnyi anyagon lett betanítva. Ezzel ellentétben a magyarlanc tanító anyaga kb. 70.000 mondatot tartalmaz, ami nagyságrendnyi különbséget jelent, és az elemzés pontosságára is kihatással van. Ugyanakkor a nemzetközi összevethetőség szemszögéből nézve az UD-sémára épülő elemzés többnyelvű vizsgálatok esetén hasznosabb lehet, mint a magyarlanc „magyarspecifikus” jegyekkel is bíró kimenete (megjegyezzük, hogy ez utóbbi szempont az UD és az e-magyar összevetésében is fennáll).

A továbbiakban megvizsgáljuk, hogyan tudunk programozói tudás nélkül is adatokat gyűjteni az elemzett fájlokból.

Első lépésben válasszunk ki egy nekünk szimpatikus szöveget! Ez lehet akár saját írásunk, akár az internetről gyűjtött szöveg, lényeg, hogy szöveges formátumban (txt) álljon rendelkezésünkre. Amennyiben egy internetes oldal tartalmát szeretnénk feldolgozni, illetve szöveges formátumban elmenteni, segítséget nyújthat a boilerpipe nevű eszköz. A https://boilerpipe-web.appspot.com oldalon elérhető eszköz megfelelő sorába illesszük be a letölteni kívánt oldal linkjét (legyen ez a példánkban a https://www.delmagyar.hu/szeged_hirek/kilometerekben_keszul_a_bejgli_-_az_elmaradhatatlan_klasszikus_karacsonyi_edesseget_kostoltuk/2583243 link), az Output Mode-ot állítsuk Plain textre, azaz sima szöveges állományra, majd nyomjunk az Extract gombra (2. ábra)! Ha a szöveges állomány még tartalmaz a weboldalról más fölösleges részleteket, kísérletezzünk azzal, hogy az Extractort is megváltoztatjuk, például LargestContentExtractorra vagy KeepEverythingExtractorra. Az 1. ábrán is látszik, hogy a weboldal eredetileg tartalmazott egy videót is, azonban a boilerpipe ezt nem exportálta szövegként.

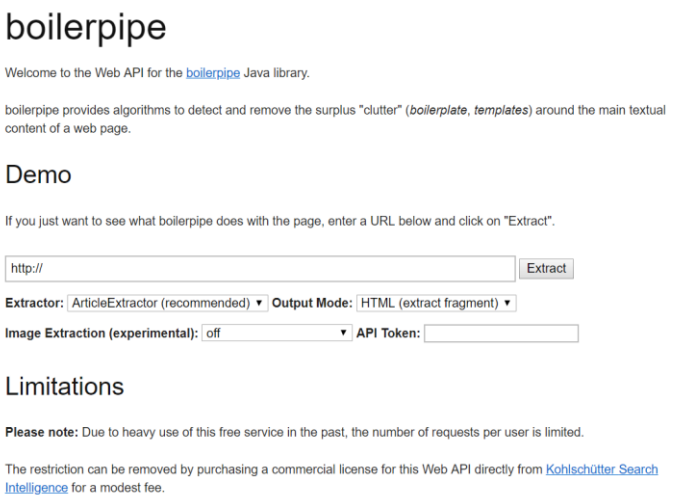
A bejgliket a Z. Nagy Cukrászdából, a Sugar & Candyből, az A Cappellából, a Reök Kézműves Cukrászda és Kávéházból, valamint a Lidlből és a Tescóból hoztuk.

A megjelenés alapján a négy cukrászdai termék átment a teszten. A Lidl bejglije méretével és kinézetével is kilógott a sorból. Az édesség az áruházláncnál 295 grammos és lapos. Viszont akciós és olcsó, 499 helyett mindössze 349 forint. A burritóhoz vagy kiflihez hasonló bejgliben ránézésre valóban sok a dió, de az ízén ez nem érződik.



A Tescóban árult édességet nem is merik bejglinek nevezni, hiszen az annak előállítására és minőségi követelményeire vonatkozó szabályokat a Magyar Élelmiszerkönyv tartalmazza, pontosan meg van határozva, milyen anyagokat lehet felhasználni a készítésükhöz, az elkészült termékeknek milyen kémiai, fizikai és érzékszervi tulajdonságokkal kell rendelkezniük. Ezeknek a tescós édesség nem felel meg, ezért omlós diós tekercs néven árulják, 400 grammot 499 forintért. Erre sem érdemes túl sok szót vesztegetni, de ebben legalább érződik a dió íze, viszont élvezhetetlenül száraz.

1. ábra: Egy online megjelent cikk



2. ábra: A boilerpipe online kezelőfelülete

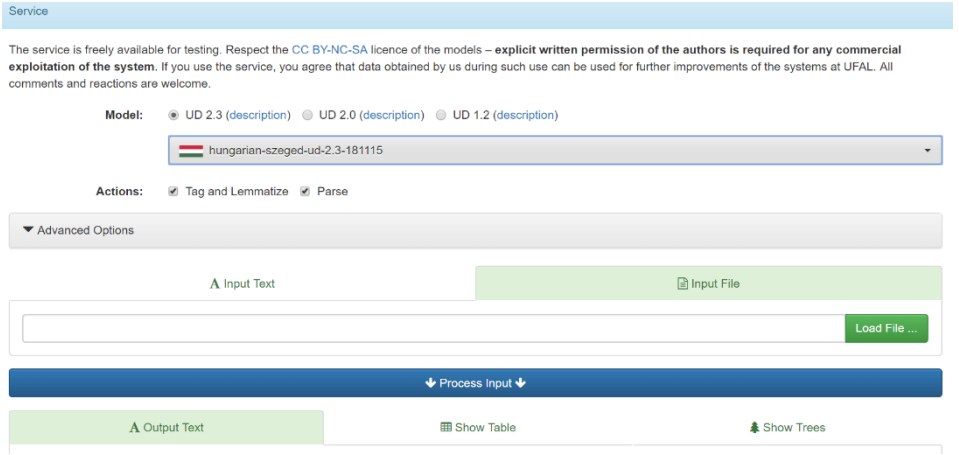
A kinyert tartalom egy részlete:

A bejgliket a Z. Nagy Cukrászdából, a Sugar & Candyből, az A Cappellából, a Reök Kézműves Cukrászda és Kávéházból, valamint a Lidlből és a Tescóból hoztuk.
A megjelenés alapján a négy cukrászdai termék átment a teszten. A Lidl bejglije méretével és kinézetével is kilógott a sorból. Az édesség az áruházláncnál 295 grammos és lapos. Viszont akciós és olcsó, 499 helyett mindössze 349 forint. A burritóhoz vagy kiflihez hasonló bejgliben ránézésre valóban sok a dió, de az ízén ez nem érződik.
A Tescóban árult édességet nem is merik bejglinek nevezni, hiszen az annak előállítására és minőségi követelményeire vonatkozó szabályokat a Magyar Élelmiszerkönyv tartalmazza, pontosan meg van határozva, milyen anyagokat lehet felhasználni a készítésükhöz, az elkészült termékeknek milyen kémiai, fizikai és érzékszervi tulajdonságokkal kell rendelkezniük. Ezeknek a tescós édesség nem felel meg, ezért omlós diós tekercs néven árulják, 400 grammot 499 forintért. Erre sem érdemes túl sok szót vesztegetni, de ebben legalább érződik a dió íze, viszont élvezhetetlenül száraz.

Amennyiben meg vagyunk elégedve a kinyert szöveges tartalommal, másoljuk ki a szöveget, és illesszük be egy szövegszerkesztőbe (Notepad vagy akár Microsoft Word), és szöveges állományként (txt) mentsük el!

Következő lépésként a mentett szöveget morfológiai és szintaktikai elemzésnek vetjük alá. Ehhez most a Universal Dependencies formalizmusra épülő UDPipe nevű eszközt használjuk fel, mely a http://lindat.mff.cuni.cz/services/udpipe/ oldalon érhető el (3. ábra). Először is minden más beállítást változatlanul hagyva válasszuk ki a magyar nyelvet, majd az Input file fülre kattintva a Load file gombbal válasszuk ki az előbbiekben elmentett txt fájlunkat! Ezután nyomjunk a Process input gombra! A Save output file gombra kattintva el tudjuk menteni az elemzett fájlt (4. ábra).

Keressük meg a fájlt a gépünkön, és szövegfájlként nyissuk meg például Notepadben! A teljes szöveg kimásolása után illesszük be az egészet egy üres Excel-munkafüzetbe (5. ábra)! Látjuk, hogy az eredeti szövegszavak a B oszlopban jelennek meg, továbbá ezek szótövesített alakjai a C oszlopot foglalják el, majd a D oszlop tartalmazza a szavak szófaját, az F az egyéb morfológiai jegyeket (például szám, személy, igeidő), végül a G és H oszlopok a függőségi elemzés



3. ábra: A UDPipe online kezelőfelülete

```
# sent_id = 2
# text = A megjelenés alapján a négy cukrászdai termék átment a teszten.
1   A   a   DET   _   Definite=Def|PronType=Art   2   det   _   _
2   megjelenés   megjelenés   NOUN   _   Case=Nom|Number=Sing|Number[psed]=None|Number[psor]=None|Person[psor]=None   3   nmod:att   _   _
3   alapján   alap   NOUN   _   Case=Sup|Number=Sing|Number[psed]=None|Number[psor]=Sing|Person[psor]=3   8   nmod:obl   _   _
4   a   a   DET   _   Definite=Def|PronType=Art   7   det   _   _
5   négy   négy   NUM   _   Case=Nom|Number=Sing|NumType=Card   6   amod:att   _   _
6   cukrászdai   cukrászdai   ADJ   _   Case=Nom|Degree=Pos|Number=Sing   7   amod:att   _   _
7   termék   termék   NOUN   _   Case=Nom|Number=Sing   8   nsubj   _   _
8   átment   átmegy   VERB   _   Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act   0   root   _   _
9   a   a   DET   _   Definite=Def|PronType=Art   10   det   _   _
10   teszten   teszt   NOUN   _   Case=Sup|Number=Sing|Number[psed]=None|Number[psor]=None|Person[psor]=None   8   nmod:obl   _   SpaceAfter=No
11   .   .   PUNCT   _   _   8   punct   _   _
```

4. ábra: A morfológiailag és szintaktikailag elemzett szöveg egy részlete

# sent_id = 12									
# text = A megjelenés alapján a négy cukrászdai termék átment a teszten.									
1	A	a	DET	_	Definite=Def|PronType=Art	2	det	_	_
2	megjelenés	megjelenés	NOUN	_	Case=Nom|Number=Sing|Number[psed]=	3	nmod:att	_	_
3	alapján	alap	NOUN	_	Case=Sup|Number=Sing|Number[psed]=N	8	nmod:obl	_	_
4	a	a	DET	_	Definite=Def|PronType=Art	7	det	_	_
5	négy	négy	NUM	_	Case=Nom|Number=Sing|NumType=Card	6	amod:att	_	_
6	cukrászdai	cukrászdai	ADJ	_	Case=Nom|Degree=Pos|Number=Sing	7	amod:att	_	_
7	termék	termék	NOUN	_	Case=Nom|Number=Sing	8	nsubj	_	_
8	átment	átmegy	VERB	_	Definite=Ind|Mood=Ind|Number=Sing|Pe	0	root	_	_
9	a	a	DET	_	Definite=Def|PronType=Art	10	det	_	_
10	teszten	teszt	NOUN	_	Case=Sup|Number=Sing|Number[psed]=N	8	nmod:obl	_	SpaceAfter=No
11	.	.	PUNCT	_	_	8	punct	_	_

5. ábra: A morfológiailag és szintaktikailag elemzett szöveg egy részlete Excelben megjelenítve

részleteit takarják. Excel-szűrésekkel egyszerű statisztikai adatokat is tudunk gyűjteni, például: az adott szófajok aránya a szövegben, a leggyakoribb főnevek, amelyek alanyként szerepelnek a szövegben, tulajdonnevek a szövegben stb.

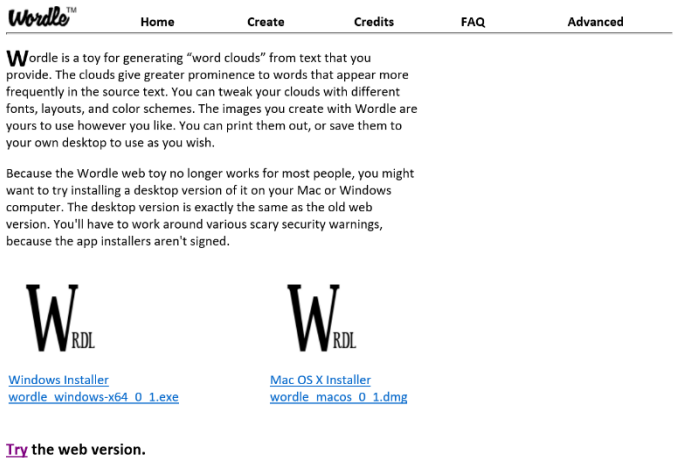
Tegyük fel, hogy a példában a szövegben megjelenő leggyakoribb főneveket szeretnénk egy szófelhő segítségével vizualizálni! Ehhez először is kapcsoljuk be

	A	B	C	D	E	F	G	H	I	J	K
1	# newd										
6	2	hírei	hír	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	5	nsubj	_	_	
8	4	Kilométerekben	Kilométerek	NOUN	_	Case=Ine|Number=Plur	5	nmod:obl	_	_	
18	14	édességet	édesség	NOUN	_	Case=Acc|Number=Sing	15	obj	_	_	
20	16	Kilométerekben	Kilométerek	NOUN	_	Case=Ine|Number=Plur	17	nmod:obl	_	_	
30	26	édességet	édesség	NOUN	_	Case=Acc|Number=Sing|Number[psed]=N	27	obj	_	_	
50	1	Diósból	Diós	NOUN	_	Case=Ela|Number=Sing|Number[psed]=N	2	nmod:obl	_	_	
52	3	hatot	hat	NOUN	_	Case=Acc|Number=Sing|Number[psed]=N	4	obj	_	_	
51	3	mindenkit	mindenk	NOUN	_	Case=Acc|Number=Sing|Number[psed]=N	4	obj	_	_	
56	8	ízével	íz	NOUN	_	Case=Ins|Number=Sing|Number[psed]=N	4	nmod:obl	_	SpaceAfter=No	
70	12	árával	ár	NOUN	_	Case=Ins|Number=Sing|Number[psor]=Sir	8	conj	_	_	
78	3	halászlé	halászlé	NOUN	_	Case=Nom|Number=Sing|Number[psed]=	11	obl	_	_	
89	14	asztalról	asztal	NOUN	_	Case=Del|Number=Sing|Number[psed]=N	15	nmod:obl	_	_	
96	2	cukrászdák	cukrászda	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	16	obl	_	SpaceAfter=No	
98	4	pékségek	pékség	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	2	conj	_	_	
01	7	bevásárlóközpontok	bevásárlóközpont	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	13	nmod:att	_	SpaceAfter=No	
03	9	áruházak	áruház	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	7	conj	_	_	
06	12	kisboltok	kisbolt	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	7	conj	_	_	
07	13	polcai	polc	NOUN	_	Case=Nom|Number=Plur|Number[psed]=	16	nsubj	_	_	
09	15	bejglitől	bejgli	NOUN	_	Case=Abl|Number=Sing	16	nmod:obl	_	_	

6. ábra: Az elemzett szövegből leszűrt főnevek

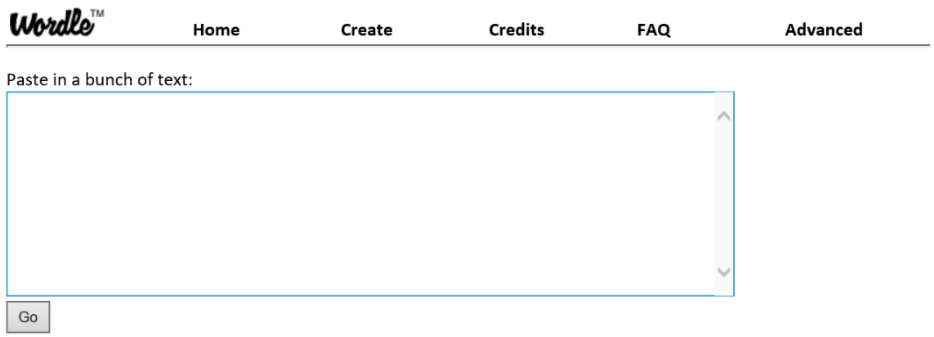
az Excel szűrő funkcióját, és a D oszlopból gyűjtsük ki a főneveket (NOUN) (6. ábra)! A szűrt sorokban jelöljük ki a C oszlopot (feltételezve, hogy a szótövesítés utáni alakok gyakorisága érdekel minket, tehát a *bejglit*, *bejglivel* stb. alakokat egyként (*bejgli*) szeretnénk kezelni). Az így kapott szólistát fogjuk vizualizálni a Wordle program segítségével.

Ehhez nyissuk meg a http://www.wordle.net oldalt, itt kattintsunk a Try the web version opcióra (7. ábra)! (Ha nem működik, akkor érdemes letölteni a programnak az operációs rendszerünknek megfelelő asztali verzióját, majd feltelepíteni azt, az utasításokat követve.) Amennyiben működik a webes változat, illesszük be az előzőekben az Excelből leszűrt főnévlistát, majd kattintsunk a Go gombra (8. ábra)! Eredményül egy szófelhőt kell kapnunk, melyen a betűméret



7. ábra: A Wordle online kezelőfelülete

jelöli az előfordulási gyakoriságot, tehát minél nagyobb betűkkel jelenik meg egy szó, annál gyakrabban fordult elő szövegünkben (9. ábra). A mi példánkban a *bejgli* szó tűnik a leggyakoribbnak, de az *édesség*, *forint*, *íz* és *diós* szavak is sokszor fordultak elő.



8. ábra: Szöveg beillesztése a Wordle-be

9. ábra: Az újságcikk leggyakoribb főnevei, a Wordle segítségével megjelenítve

A Font, Layout és Color menüpontokban igény szerint szabadon változtathatjuk az ábra színeit, betűtípusát, a Language menüpontban pedig be tudjuk állítani, hogy a leggyakoribb nyelvtani szavakat (az úgynevezett stopszavakat, mint például *és*, *vagy*, *a*, *ez*, *az*, *van*…) figyelembe vegye-e a program. Lehetőségünk van a szófelhő elmentésére és kinyomtatására is.

7. Összegzés

E tanulmányban röviden bemutattuk a korpuszok jelentőségét a nyelvészeti kutatásban, valamint ismertettünk néhány önálló, programozási tudást nem igénylő módszert, melyek segítségével a korpuszokból adatokat tudunk gyűjteni, illetve azokat elemezni. A tanulmánynak nem lehetett célja a teljes részletességre törekvés sem a módszerek, sem a korpuszok ismertetésekor, azonban az érdeklődő olvasó számára az alábbiakban szeretnénk néhány további lehetséges irányt felvázolni.

A korpusznyelvészetről részletes áttekintést nyújt Szirmai Monika könyve (Szirmai 2006). Az elemzett korpuszokban való kereséshez, különös tekintettel a Magyar Nemzeti Szövegtárra, Sass Bálint e kötetbeli tanulmánya mutat be különböző módszereket (Sass 2019), illetve a Nemzeti Korpuszportálon összegyűjtött korpuszokban is lehetséges adatokat keresni. A történeti korpuszokról Simon Eszter, a gyermeknyelvi korpuszokról Babarczy Anna tanulmányában olvashatunk részletesen (Simon 2019, Babarczy 2019). Végül egy további alkalmazást is szeretnénk az olvasó figyelmébe ajánlani: a TANIT online szolgáltatás a magyarlanc elemzéseire építve képes a szövegre jellemző alapvető statisztikai adatokat automatikusan összegyűjteni (lásd Péter 2019). Akit pedig mélyebben érdekel a programozás, Hammond *Java for Linguists* című könyvéből elsajátíthatja a nyelvészeti kutatáshoz szükséges programozás alapjait (Hammond 2002).

Irodalom

Babarczy A. 2019. Gyermeknyelvi korpuszok és erőforrások. In: Sulyok H., Juhász V., Erdei T. (szerk.). *Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért. HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Szeged: SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék.

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton and Co.

Csendes D., Csirik J., Gyimóthy T., Kocsor A. 2005. The Szeged Treebank. In: Matoušek, V. et al. (szerk.). *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD 2005)*. Berlin, Heidelberg: Springer-Verlag. 123–131.

Durst P., Szabó M. K., Vincze V., Zsibrita J. 2013. A HunLearner magyar tanulói korpusz fejlesztése és várható hozadékai. *THL2: A magyar nyelv és kultúra tanításának szakfolyóirata* 9/1–2. 28–41.

Hammond, M. 2002. *Programming for linguists: Java™ technology for language researchers*. Oxford: Blackwell.

Mittelholcz I. 2019. Bevezetés az `e-magyar` programcsomag használatába. In: Sulyok H., Juhász V., Erdei T. (szerk.). *Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért*. *HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Szeged: SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék.

Oravecz Cs., Váradi T., Sass B. 2014. The Hungarian Gigaword Corpus. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC2014)*. Reykjavík: European Language Resources Association.

Péter R. 2019. A big data kihívás a bölcsészettudományokban: néhány digitális bölcsészeti kutatási eszköz bemutatása. In: Sulyok H., Juhász V., Erdei T. (szerk.). *Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért*. *HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Szeged: SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék.

Sass B. 2019. Keresés korpuszban 2: így kerestek ti. In: Sulyok H., Juhász V., Erdei T. (szerk.). *Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért*. *HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Szeged: SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék.

Simon E. 2019. Magyar nyelvű történeti korpuszok. In: Sulyok H., Juhász V., Erdei T. (szerk.). *Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért*. *HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban*. Szeged: SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék.

Straka, M., Straková, J. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver: Association for Computational Linguistics. 88–99.

Szarvas Gy., Farkas R., Kocsor A. 2006. A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: *Discovery Science 2006*. Berlin, Heidelberg: Springer-Verlag 267–278.

Szirmai M. 2006. *Bevezetés a korpusznyelvészetbe*. *A korpusznyelvészet alkalmazása az anyanyelv és az idegen nyelv tanulásában és tanításában*. Budapest: Tinta Kiadó.

Vincze V. 2018. A Miskolc Jogi Korpusz nyelvi jellemzői. In: Szabó M., Vinnai E. (szerk.). *A törvény szavai: Az OTKA-112172 kutatási zárókonferencia anyaga*. Miskolc: Bíbor Kiadó. 9–36.

Zsibrita J., Vincze V., Farkas R. 2013. magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: *Proceedings of RANLP 2013*. Hissar: Association for Computational Linguistics. 763–771.

Keresés korpuszban 2: így kerestek ti[1]

Sass Bálint
tudományos munkatárs
MTA Nyelvtudományi Intézet, Nyelvtechnológiai Kutatócsoport
ELTE Bölcsészettudományi Kar
sass.balint@nytud.mta.hu

2001 óta foglalkozom számítógépes nyelvészettel, korpuszokkal. Programtervező matematikusként végeztem 2003-ban, a PhD-mat 2011-ben védtem meg, témája egy korpuszból igei szerkezeteket automatikusan kinyerő, szótárkészítést segítő, korpuszvezérelt algoritmus volt. Ennek kapcsán 2010-ben jelent meg a Magyar igei szerkezetek szótár. Fontos szakterületem a korpuszlekérdezés módszereinek vizsgálata, oktatása. Az MTA Nyelvtudományi Intézetében számos korpusszal dolgoztam, részt vettem az egymilliárd szónyi mai magyar szöveget tartalmazó Magyar Nemzeti Szövegtár összeállításában, a helyesiras.mta.hu portál létrehozásában, foglalkoztam a magyar Braille-rövidírás megújításával. Mostanában az e-magyar nyelvelemző rendszert gondozom, és legóból építek igei szerkezeteket.
http://www.nytud.hu/depts/corpus/Sass_Balint.html

1. Bevezetés

Korábbi írásaimban megfogalmaztam a Nemzeti Korpuszportál koncepcióját (Sass 2016), valamint bemutattam a korpuszokban való keresés alapvető módszereit (Sass 2017). A Nemzeti Korpuszportál célja a szóalapú online keresővel rendelkező magyar nyelvű korpuszok összegyűjtése, és távlatilag az elemzőeszközök és a korpuszkeresési funkciók elérhetővé tétele minden korpusz számára. A korpuszban való keresés vonatkozásában áttekintettük a Nemzeti Korpuszportálon több esetben is alkalmazott NoSketchEngine (NoSkE) (Rychlý 2007) korpuszkezelő rendszer funkcióit, különös tekintettel a szűrésre és a gyakorisági listákra, illetve a reguláris kifejezésekre, valamint az arra épülő Corpus Query Language (CQL) formális lekérdezőnyelvre, melynek segítségével teljeskörűen feltárható egy korpusz annotációjában kódolt minden nyelvi és egyéb információ.

[1] Az Információs és Technológiai Minisztérium ÚNKP-19-4 kódszámú Új Nemzeti Kiválóság Programjának szakmai támogatásával készült.

2. Lekérdezések osztályai

A tanulmány további részében feltételezzük az említettek ismeretét. Most a Magyar Nemzeti Szövegtár kibővített változata (MNSZ2) (Orvecz et al. 2014) keresőfelületén a korpusz használói által megadott valódi CQL lekérdezésekből – mondhatjuk így: a CQL korpuszlekérdezések alkotta „korpuszból" – szemezgetünk, ezeket elemezzük. Levonjuk a tanulságokat, számba vesszük az ajánlott megoldásokat és az elkerülendő próbálkozásokat. Konkrét gyakorlati tippeket adunk a korpuszlekérdező rendszer használatával kapcsolatban.

A vizsgált lekérdezéseket az elemzésünk során három csoportra fogjuk osztani:

1. érdekes, értelmes lekérdezések
2. hibák
3. „ilyet ne"

A harmadik csoporthoz egy kis magyarázat: ide azok a lekérdezések tartoznak, amelyek kezelhetetlenül sok – rosszabb esetben teljesen haszontalan – találatot eredményeznek, ráadásul nagymértékben leterhelik a korpuszkezelő hardveres és szoftveres környezetet. Az ilyen lekérdezéseket lehetőség szerint próbáljuk meg elkerülni, általában azon a módon, hogy a lekérdezés átgondoltabb megfogalmazásával leszűkítjük, jobban specifikáljuk, jobban meghatározzuk, hogy pontosan mire is vagyunk kíváncsiak.

3. Elemzendő példák

Tekintsük át tehát az alábbi tíz példát, CQL lekérdezést!

a) `"tudatjuk" "mindazokkal"`

b) `[lemma="felkap"] [lemma="a"] [lemma="víz"]`

c) `[word="."]`

d) `[lemma = "k[K]andeláber"]`

e) `ama i*`

f) `[word="\."] [word="[Mm]indig"] [word="\."]`

g)
```
[msd="Det.*"] [msd="FN.PSe2.*"] [lemma="fog"]
  [word=".*ni" & pos="V.*"]
```

h) `".*"`

i) `[][][][]*`

j) `[word = "elé"] [word = "a.?"] [word = ".+n(a|e)k"]`

Mielőtt a következő fejezetben található magyarázatokat, megjegyzéseket elolvassuk, érdemes elgondolkodni azon, hogy az egyes lekérdezéseknek vajon mi a tartalma és célja, és melyik lekérdezést, melyik fenti csoportba sorolnánk.

4. Elemzés

4.1. a) `"tudatjuk" "mindazokkal"`

Első példánk egy hasznos, értelmes, ugyanakkor nagyon egyszerű lekérdezés. Két egymást közvetlenül követő szóalakra (*tudatjuk* és *mindazokkal*) kérdez rá. Kihasználja a NoSkE azon lehetőségét, hogy ha default attribútumot adunk meg a felületen, akkor a CQL lekérdezésbe nem kell beleírnunk ezt az attribútumot, hanem csak egyszerűen idézőjelbe kell tenni a keresett szavakat, illetve a szavakat megadó reguláris kifejezéseket. (Tovább azonban nem egyszerűsíthető a lekérdezés, az idézőjelek nem elhagyhatók, mindenképpen kellenek.) Így, ha default attribútumnak a `word`öt (szóalak) állítjuk be, akkor

```
[word="tudatjuk"] [word="mindazokkal"]
```

helyett írhatjuk egyszerűen a példában látottakat. A lekérdezés segítségével természetesen jó eséllyel gyászjelentések szövegrészleteit fogjuk eredményül kapni, ez is lehetett a lekérdezés tényleges célja erre a nagyon jellegzetes fordulatra való kereséssel.

4.2. b) `[lemma="felkap"] [lemma="a"] [lemma="víz"]`

A példa a `lemma` (szótő) attribútum segítségével három egymást közvetlenül követő szótőre kérdez rá. Nyilván a *felkapja a vizet* szólás előfordulásaira volt kíváncsi a lekérdező, azaz ez a példa is az 1. osztályba tartozik.

Elegendő lett volna a szótőre keresést csupán az igére korlátozni, mivel a szólás másik két eleme fix, így:

```
[lemma="felkap"] [word="a"] [word="vizet"]
```

Vegyük észre ugyanakkor, hogy ez a lekérdezés nem fogja megtalálni e szólás összes előfordulását, mégpedig a különféle lehetséges szórendek, valamint az esetlegesen elváló igekötő miatt. Érdemes elgondolkodni azon, hogyan tudjuk ezeket a problémákat általánosságban kezelni, valamint hogyan tudunk olyan lekérdezést alkotni, amely lehetőséget ad a *víz* elem példányaiból álló gyakorisági lista elkészítésére, lehetővé téve a fenti állítás ellenőrzését, miszerint ez az elem fix.

Ha szóalak szerinti gyakorisági listát készítünk a találatokból, megtudhatjuk, hogy ennek a szólásnak ebben a szórendben mely ragozott alakjai a leggyakoribbak: az első a *felkapja a vizet*, a második a *felkapta a vizet*, a kettő együtt a találatoknak több mint a felét teszi ki.

4.3. c) `[word="."]`

Mire szolgálhat ez a lekérdezés? Tudjuk, hogy a CQL-lekérdezésekben egy token (azaz szó vagy írásjel) adott attribútumára vonatkozó feltételt egy idézőjelek között megadott reguláris kifejezés testesíti meg az alábbi formában:

```
[attribútum="regkif"]
```

A reguláris kifejezésekben a pont (**.**) speciális jelentéssel bír, azt jelenti, hogy azon a helyen tetszőleges karakter állhat. Így az egyetlen pontból álló reguláris kifejezés az egy darab tetszőleges karakterből álló tokenekre kérdez rá. Ebben az esetben az eredményhalmaz egyrészt roppant méretű, az egymilliárd szavas MNSZ2 esetén akár százmilliós nagyságrendű, mivel az írott magyar szövegnek akár 30%-át is alkothatják az egykarakteres tokenek. Másrészt nagyon heterogén az eredményhalmaz, a hasznos közös tulajdonsággal nem nagyon rendelkező különféle egybetűs szavakon (*a, ő, ó, s* stb.) kívül tartalmazza a számos egykarakteres írásjel példányait is. Nem könnyű rájönni, hogy mi lehetett itt a kérdező szándéka, az mindenesetre biztos, hogy ez egy „ilyet ne” típusú lekérdezés.

Lehetséges, hogy egyszerűen a mondatvégi pont írásjelre szeretett volna rákérdezni itt a korpusz használója, de nem volt tisztában azzal, hogy az attribútumok értéke reguláris kifejezésként értelmeződik, és ott a pont speciális karakter. Ha egy speciális karaktert „eredetiben”, azaz speciális jelentésétől mentesítve szeretnénk használni, akkor „escape”-elni kell a visszaper karakter segítségével, ekkor tehát a helyes lekérdezés a következő:

```
[word="\."]
```

4.4. d) `[lemma = "k[K]andeláber"]`

A szándék itt nyilvánvalóan az volt, hogy a *kandeláber* szótőre keresve a szó összes ragozott alakját is megkapjuk úgy, hogy a szó kisbetűvel és nagybetűvel írt verzióit is számításba vesszük. Valóban, ha szükségünk van a kisbetűs és a nagybetűs változatra is, akkor ezt explicite meg kell adnunk. Erre használható a reguláris kifejezésekben a szögletes zárójel, ami egy karakterlistát tartalmazva azt

jelenti, hogy a megadott karakterlistából pontosan az egyik karakter szerepelhet az adott ponton. A

```
[eö]
```

regkif jelentése tehát *e* vagy *ö*, ennek megfelelően a

```
tejf[eö]l
```

jelentése *tejfel* vagy *tejföl*. A kisbetű/nagybetű – esetünkben kis *k* és nagy *K* – vagylagos megadása tehát a következő módon lehetséges:

```
[Kk]
```

Azonban a d) példában a szögletes zárójel csak egy karaktert tartalmaz, ez pont olyan, mintha egyáltalán nem használnánk szögletes zárójelet, azaz a példa azonos értelmű az alábbi – találatot értelemszerűen nem adó – lekérdezéssel:

```
[lemma = "kKandeláber"]
```

A lekérdezés tehát hibás, a 2. osztályba tartozik. A felhasználó szándékának megfelelő lekérdezés a következő lehetett volna:

```
[lemma = "[Kk]andeláber"]
```

Fontos látni a különbséget a kétféle `[]` között: a belső egy reguláris kifejezés részeként karakterek vagylagosságát fejezi ki, a külső az egy tokenre vonatkozó feltétel(eke)t tartalmazza a CQL elemeként.

Megjegyezzük, hogy a tulajdonnevek kivételével a szavak szótöve alapesetben kisbetűs, akkor is, ha egy mondat elején álló nagybetűs szó szótövéről van szó. Azaz itt használható lett volna egyszerűen a következő lekérdezés:

```
[lemma = "kandeláber"]
```

4.5. e) `ama i*`

A lekérdezés egyértelműen hibás, ez már onnan látszik, hogy nem tartalmaz idézőjelet. Ezt javítva

```
"ama" "i*"
```

is rejtélyes marad a lekérdezés célja. Az *ama* szó után következő tetszőleges számú *i* betűből álló szóra vonatkozik a lekérdezés, aminek nehéz hasznos funkciót tulajdonítani. Utolsó ötletként az *ama*-t követő *i*-vel kezdődő szavakra vonatkozó, még mindig nem túl mélyenszántó lekérdezés a következő lenne:

```
"ama" "i.*"
```

Mivel a reguláris kifejezésekben a csillag (`*`) operátor jelentése a „bárhány” (0 vagy több) a megelőző elemből, a tetszőleges számú tetszőleges karakter `.*` formában adandó meg.

4.6. f) `[word="\."] [word="[Mm]indig"] [word="\."]`

Értelmes lekérdezés. Egy mondatvégi pontot követő *mindig* szót keres úgy, hogy azt ismét egy mondatvégi pont kövesse. Azaz egy egy szóból álló mondat megtalálása a cél. A pont karakterek helyesen „escape”-elve vannak (vö: 4.3. rész), a *mindig* szó elején kis- és nagybetűt is megenged a lekérdezés, vélhetően általában fölöslegesen, mert a mondat eleji helyzet miatt lényegében mindig nagybetűs lesz.

Mire szolgálhat? A mondandó végén nyomatékosításként odatett egyszavas *Mindig*. mondatokat szeretné megkapni. Az írásjelek a korpuszban külön tokenek, ezért a lekérdezés fenti formája helyes.

Az MNSZ2 keresőjébe beírt lekérdezések között több hasonló felépítésű is szerepel (*Sokszor., Teljesen.* stb.), a felhasználó valószínűleg az ilyen formájú mondatokról szándékozott általános tanulságokat levonni. Ilyen esetben – szem előtt tartva, hogy lehetőleg az összes releváns korpuszadatra építsük a vizsgálatainkat – érdemes lehet egy általánosabb lekérdezéssel kezdeni, abból gyakorisági listát csinálni és azon vizsgálódni, hogy ne csak az intuitíve felmerülő példákkal dolgozzunk, hanem az összes rendelkezésre álló valós nyelvi adattal.

A javaslat itt konkrétan a

```
"\." [] "\."
```

lekérdezés lenne, amely a vizsgált pozícióban tetszőleges szót megenged (azáltal, hogy az egy tokenre vonatkozó szögletes zárójelpár belsejében nem köt ki feltételt). A találatok között a leggyakoribbak a *Rip. Köszönöm. Köszönjük. Nem. Igen. Ennyi.*, aztán a gyakorisági listán lejjebb következnek a vizsgálat szempontjából vélhetően relevánsabb megnyilatkozások: *Sajnos. Mindegy. Szép. Talán.* stb.

4.7. g) `[msd="Det.*"] [msd="FN.PSe2.*"] [lemma="fog"] [word=".*ni" & pos="V.*"]`

Ebben a fejezetben két olyan lekérdezést vizsgálunk, amelyek ugyanazt a problémát példázzák. Elsőre úgy tűnik, hogy mindkét lekérdezés rendben van, az első az `msd` (morfológiai kód) attribútum segítségével névelős egyes szám máso-

dik személyű birtokos főneveket keres (pl.: *a lovad, az árnyékodba*), a második pedig a *fog* + főnévi igenév szerkezeteket a *-ni* végződés és az igei („ige” szófaj-kóddal kezdődő) morfológiai kód alapján.

A hiba mindkét esetben ott van, hogy nem megfelelő kódokat használtunk. Az MNSZ2-ben (a v2.0.2–v2.0.5 verziókban) a határozott névelő kódja nem `Det`, hanem `DET`, a szófajok kódjai magyar rövidítéssel szerepelnek, azaz az ige nem `V`, hanem `IGE`, és a morfológiai kódot tartalmazó attribútum elnevezése pedig nem `pos`, hanem `msd`.

Ha nem vagyunk biztosak a kódokban, a legegyszerűbb, ha rákeresünk egy olyan konkrét szó(kapcsolat)ra, amilyen jellegűre kíváncsiak vagyunk:

```
"a" "neved.+"
```

(ahol a .+ segítségével a ragozott alakokat is megengedtük), illetve

```
"fog" "csinálni"
```

majd a megjelenítésben bekapcsoljuk az `msd` attribútumot, és megnézzük, hogy milyen kódokat kell használnunk.

4.8. h) `".*"`

Nagyon egyszerű, de annál kártékonyabb lekérdezés. A korábbiakból tudjuk, hogy reguláris kifejezésben a pont (`.`) tetszőleges karakterre illeszkedik, a csillag (`*`) operátor jelentése pedig az, hogy a megelőző karakterből „bárhány” (0 vagy több) darab. Így a `.*` jelentése: bárhány egymást követő tetszőleges karakter (beleértve a nulla darabot is, azaz az üres szót is, ami pusztán elméleti lehetőség, a korpuszokban nem fordul elő).

A 4.7. részben láttuk, hogy nagyobb regkif részeként milyen hasznos motívum a `.*`, mert a segítségével tudjuk például attribútumértékek elejét vagy végét megadni. Így önmagában viszont a tetszőleges szóra való keresést jelenti, ami az egymilliárd szavas MNSZ2 esetében elvben egymilliárd találatot eredményez, sorra a korpusz összes szavát, a gyakorlatban viszont a túlzott erőforrásigény miatt nem fog lefutni. Ez a lekérdezés az „ilyet ne” csoportba tartozik.

4.9. i) `[][][][]*`

A 4.6. részben láttuk, hogy a `[]` a „tetszőleges szó” megadásának egyik módja. (Azt is láttuk az imént, hogy a `".*"` szintén ezt jelenti. Valójában mindkettő a

```
[word=".*"]
```

lekérdezés egyszerűsített változata, az első a feltétel elhagyása révén, a második default attribútum használata révén.) A szimpla

```
[][][][]
```

lekérdezés is bőven az „ilyet ne” kategóriába esik, a 4.8. részben látott példához hasonlóan egymilliárd találatot eredményez, de a `*` segítségével még sokkal rosszabb lesz a helyzet. Ez az operátor az utolsó *token* többszörözését engedi meg, a lekérdezés ezáltal a korpusz legalább háromtokenes egységeire kérdez rá. Ennek megfelel az 1-2-3. token, az 1-2-3-4. token, az 1-2-3-4-5. token stb. egészen addig, hogy 1-2-3-...-1000000000. token, aztán a 2-3-4. token, a 2-3-4-5. token, a 2-3-4-5-6. szó – és így tovább. Ez az egymilliárd szavas korpusz esetén nagyságrendileg $10^9 \cdot 10^9 / 2 = 5 \cdot 10^{17}$ darab találatot eredményez, ami találatonként csupán 10 byte-tal számolva is 5 millió terabyte adat lenne. Ez a lekérdezés tehát az „ilyet ne” kategória minősített esete.

4.10. j) `[word = "elé"] [word = "a.?"] [word = ".+n(a|e)k"]`

Utolsó példánkban az *elé* szót követő határozott névelős *-nAk* ragos szóra keresünk. A névelőt meg lehetne fogalmazni a következő módon is:

```
[word = "az?"]
```

illetve, ha a határozatlan névelőt is hozzávesszük, akkor:

```
[word = "egy|az?"]
```

a *-nAk* ragos szót pedig így:

```
[word = ".+n[ae]k"]
```

esetleg igény szerint főnévre leszűkítve

```
[word = ".+n[ae]k" & msd="FN.*"]
```

– bár erre nem nagyon van szükség, mert névelő után egyébként is jó eséllyel főnevet (névszót) fogunk kapni.

Ez egy hasznos, értelmes lekérdezés, olyan, ami láthatólag egy konkrét nyelvi jelenségre igyekszik rákérdezni. Arról a jelenségről van szó, amikor az igekötő mintha egyben névutóként is funkcionálna, ráadásul a hozzá kapcsolódó *-nAk*-ragos szó előtt megjelenve.

Erre a ritka szerkezetre (konkrétan az *elé* igekötővel/névutóval) az MNSZ-ben is csak néhány példa van: *Hogy néz elé a tárgyalássorozatnak?* vagy *Nem éltek elé a gyerekeknek olyan élet és emberi méltóság iránti tiszteletet, amely beépült volna a lelkükbe*.

A lekérdezésre jóval több találatot kapunk, de ezek közül ki kell szűrnünk azokat, amelyek nem a kívánt jelenséget testesítik meg, hanem csak „véletlenül" kerültek a kívánt alakú szavak egymás mellé: *Kilin elé a rendelésnek megfelelően a három lágy tojás került és a gyógyvíz*. A jó példáknál általában ige előzi meg a háromelemű szókapcsolatot, érdemes lehet erre szűrni.

Végül megjegyezzük, hogy az ilyenfajta, több szóból álló kifejezéseket nagy eséllyel gyorsabb szűréssel keresni, mégpedig úgy, hogy először a legspecifikusabb szóra keresünk (*elé*), majd a találatokat szűrjük a következő legspecifikusabb szóval (itt a 2-es pozícióban, azaz a találati szótól kettővel jobbra elhelyezkedő *-nAk* ragos szóval), végül a harmadik elemmel.

5. Befejezés

Reméljük, hogy a fenti elemzések közelebb hozták az olvasóhoz a CQL-t és a reguláris kifejezéseket, és a jövőben bátran fogja alkalmazni ezeket az eszközöket korpuszokban való keresés során. Bízunk benne, hogy a korábbiakban megfogalmazott alapvetést (Sass 2017) hasznos tippekkel tudtuk kiegészíteni, és hozzájárultunk ahhoz, hogy az említett eszközök használata készséggé váljon. A jövőben elkerülhetők az alapvető, fent elemzett hibalehetőségek, és így gyorsabbá, hatékonyabbá válik a korpuszt feltáró, adatgyűjtő munka.

Irodalom

Sass B. 2016. Nyelvészeti szövegkeresők, Nemzeti Korpuszportál. *Magyar Tudomány* 177/7. 798–808.

Sass B. 2017. Keresés korpuszban: a kibővített Magyar történeti szövegtár új keresőfelülete. In: Forgács T., Németh M., Sinkovics B., (szerk.). *A nyelvtörténeti kutatások újabb eredményei IX.* Szeged: SZTE Magyar Nyelvészeti Tanszék. 267–277.

Rychlý, P. 2007. Manatee/Bonito – A modular corpus manager. In: *Proceedings of the 1st Workshop on Recent Advances in Slavonic Natural Language Processing*. Brno: Masaryk University. 65–70.

Oravecz Cs., Váradi T., Sass B. 2014. The Hungarian Gigaword Corpus. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC2014)*. Reykjavík: European Language Resources Association.

Magyar nyelvű történeti korpuszok

Simon Eszter
tudományos főmunkatárs
MTA Nyelvtudományi Intézet, Nyelvtechnológiai és Alkalmazott Nyelvészeti Osztály
simon.eszter@nytud.mta.hu

A számítógépes nyelvészeten belül kutatási területeim közé tartozik a tulajdonnév-felismerés, a morfológiai elemzés, a korpuszépítés és -annotáció, a történeti korpuszok fejlesztése, valamint az uráli nyelvek számítógépes nyelvészeti támogatása. Számos hazai és nemzetközi projektben vettem részt, amelyek egy részében a számítógépes nyelvészeti munkálatok koordinátora is én voltam. Több egyetemen tartottam nyelvészeti tematikájú kurzusokat, emellett rendszeresen bírálok szakdolgozatokat, disszertációkat, cikkeket, absztraktokat, pályázatokat hazai és nemzetközi színtéren egyaránt.

1. Bevezetés

A nyelvi kulturális örökség elérhetővé tételében kulcsfontosságú szerep jut a nyelvtechnológiának, melynek módszereivel a kutatók egységes, következetes, nyelvi információval ellátott adatbázisokhoz juthatnak. A nyelvtörténészek és nyelvtechnológusok egyik legfontosabb együttműködési terepe a történeti korpuszok építése, melyek kiváló alapanyagot szolgáltatnak az elméleti és történeti nyelvészeti kutatásoknak. Az elmúlt évtizedekben számos történeti korpuszt fejlesztettek – elsősorban indoeurópai nyelvekre, de a magyarra is készült néhány. Időrendi sorrendben haladva ezek a következők. Az Ómagyar Korpusz (Simon 2014) tartalmazza az összes ómagyar korból fennmaradt szövegemléket és néhány középmagyar kori bibliafordítást is. A Történeti Magánéleti Korpusz (Novák et al. 2018) az ó- és középmagyar kor magánéleti nyelvi regiszteréhez közelebb álló műfajokat tartalmazza: 1772 előtti magánlevelekből és peres eljárások jegyzőkönyveiből épül fel. A Magyar Történeti Szövegtár (Csengery 2006) pedig 1772-től, vagyis az újmagyar kor kezdetétől egészen a 20. század végéig tartalmaz szövegeket.

Jelen tanulmány a Beszéd- és nyelvelemző szoftverek a versenyképességért és az esélyegyenlőségért – HunCLARIN korpuszok és nyelvtechnológiai eszközök a bölcsészet- és társadalomtudományokban címmel, 2018. október 19-én Szegeden tartott workshopon elhangzott előadásom írott változata. Előadásomban a fent említett korpuszokat és a hozzájuk tartozó lekérdezőfelületeket ismertettem, és néhány példán keresztül azt is illusztráltam,

hogy milyen kutatási kérdésekre hogyan tudunk választ kapni ezeknek az adatbázisoknak a segítségével. A tanulmány felépítése követi az előadás menetét: először a történeti korpuszok jellemzőit ismertetem a 2. fejezetben, majd a 3. fejezetben röviden áttekintem a történeti szövegek feldolgozásának kihívásait, végül a 4. fejezetben ismertetem a magyar nyelvű történeti korpuszokat, és egy vizsgálattal illusztrálom, hogy milyen kutatási kérdések megválaszolásához lehet használni ezeket az adatbázisokat.

2. A történeti korpuszok jellemzői

Egy történeti korpusz elsősorban korpusz, és mint ilyenre vonatkozik rá minden, ami általában egy korpuszról elmondható, vagyis: szövegek vagy szövegrészletek véges elektronikus gyűjteménye, amely jól körülhatárolt és nyelvészetileg releváns kritériumok alapján lett válogatva, valamint legalábbis törekszik a reprezentativitásra (Claridge 2008: 242). A hangsúly a törekvésen van, de a gyakorlatban a reprezentativitás egy mozgó célpont, ami a történeti korpuszok esetében még nehezebben érhető el, mint az általános célú modern korpuszok esetében.

Ami a történeti korpuszokat történetivé teszi, az az, hogy azzal a céllal készülnek, hogy reprezentálják egy nyelv régi állapotait, valamint hogy tanulmányozni lehessen rajtuk a nyelv változásait. Felmerül a kérdés, hogy mit értünk régi állapot alatt? Jellemzően azokat a szövegeket szokták réginek nevezni, amelyek legalább egy generációnyival visszanyúlnak a mai nyelvállapot előttre (Claridge 2008: 242).

A történeti korpuszoknak több típusát lehet elkülöníteni. A tipizálás egyik dimenziója mentén szinkrón és diakrón korpuszokat különböztetünk meg. A szinkrón történeti korpusz a nyelvnek egy múltbeli szeletét mutatja be, arról készít pillanatfelvételt. Ez a „pillanat” a történeti szövegek esetében akár egy évszázados is lehet, mint például a Century of Prose Corpus (Milíc 1990) esetében, amely az angol próza 1680 és 1780 közötti időszakát mutatja be. A diakrón korpusz ezzel szemben egy nagyobb időintervallumot ölel fel, ami a nyelv longitudinális vizsgálatára ad lehetőséget. Erre példa a Helsinki Corpus of English Texts (Rissanen et al. 1991), amely majd egy évezredet fog át (ca. 750–1710).

A korpusztipizálás egy másik dimenziója a korpusz felhasználása lehet, vagyis hogy a korpusz általános célú vagy specifikus. Az általános célú korpusz, mint amilyen a fent említett Helsinki Corpus of English Texts, a nyelvi vizsgáló-

dások széles skáláját teszi lehetővé, míg a specifikus korpuszok egy műfajra, egy szerzőre vagy – szélsőséges esetben — akár csak egy műre koncentrálnak, mint például az Electronic Beowulf[2] esetében.

Az időkeret a történeti korpuszok esetében kiemelt fontosságú. A régebbi korokból jellemzően kevesebb szöveges anyag áll a rendelkezésünkre, ezért fordulhat elő az, hogy még a szinkrón történeti korpuszok is egy évszázadot fognak át, ahogy azt láttuk fentebb. A méret pedig szorosan összefügg az időkerettel: minél nagyobb az időkeret, valószínűleg annál nagyobb lesz a korpusz.

Nem mindegy viszont, hogy melyik korban van a vizsgált időkeret. Ha egy 21. századi évet szemelünk ki időkeretnek, és az abban az évben keletkezett magyar nyelvű szövegekből akarunk korpuszt építeni, nagy valószínűséggel sokkal nagyobb korpuszt kapunk, mint ha ugyanezt egy 16. századi évvel tennénk. Általános szabályként kijelenthető, hogy a történeti korpuszok jellemzően kisebbek, mint a modernek. Ennek egyik oka, hogy a történeti szövegek feldolgozása jellemzően nagyobb kihívást jelent, mint az eleve elektronikusan keletkezett modern szövegek egyszerű letöltése. További ok, hogy a régebbi korokban egyrészt sokkal kevesebb nyelvi anyag keletkezett, másrészt pedig ezek egy része egyszerűen elveszett az idők során. Amik fennmaradtak, azok pedig sokszor még mindig csak kéziratban, papíron érhetők el, vagy ha digitalizálták is őket, akkor is általában csak képként, vagyis a bennük levő szöveges tartalom még mindig nem közvetlenül hozzáférhető.

Itt kanyarodunk vissza a reprezentativitáshoz, amely a korpuszépítés egyik legtöbbet tárgyalt kérdése, lásd például Biber (1993). Hunston (2008) definíciója szerint a reprezentativitás a korpusz és az általa reprezentált nyelv közötti viszonyt jelenti. Ez a definíció problémás, mivel körbenforgó: azt feltételezi, hogy a nyelvről tudunk dolgokat, miközben a korpuszt azért építjük, hogy megtudhassunk dolgokat a nyelvről. Talán érdemesebb úgy megközelíteni a kérdést, hogy mi az, ami biztosan nem reprezentatív. Ha valaki a mai magyar nyelvhasználat általános célú korpuszát óhajtja megépíteni, akkor csak és kizárólag sport tematikájú blogbejegyzéseket gyűjtve ezt nem fogja tudni teljesíteni. Vagy McEnery (2004) példájával élve: képzeljük el, hogy egy kutató egy telefonos dialógusrendszer fejlesztéséhez épít korpuszt. Ha ennek eléréséhez Jane Austen regényeiből szemezget, akkor biztosak lehetünk benne, hogy nem jár jó úton. A reprezentativitás kérdésével tehát érdemes óvatosan bánni, és úgy tekinteni, mint amire törekszünk, de a

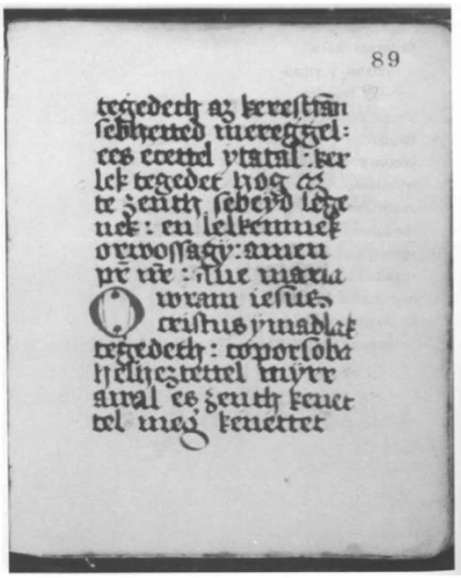
[2] http://ebeowulf.uky.edu

gyakorlatban elképzelhető, hogy sose fogjuk elérni. A történeti korpuszok esetében a reprezentativitást számos nyelven kívüli tényező is befolyásolja. Az egyik ilyen tényezőt már említettük: sok szöveg elveszett, és nem tudjuk, hogy mennyi és mi, vagyis a fennmaradt szövegek köre csak egy véletlen mintát szolgáltat a múltbeli nyelvről, nem reprezentálja a nyelv teljes változatosságát. Egy másik tényező a beszélt nyelvi adatok hiánya. A hangrögzítés technológiájának feltalálása előttről nyilvánvalóan nem lehetnek anyagaink, sőt még a kezdetleges hangrögzítő eszközök korából is csak nagyon kevés maradt fenn, azok is igen rossz minőségűek. Arra is érdemes odafigyelni, hogy a különféle műfajok, regiszterek aránya az írott nyelvváltozaton belül is kiegyensúlyozatlan – sok a vallási irodalom, kevés az informális, a sajtó, a tudományos stílus, aminek hátterében különféle szociopolitikai okok húzódnak. Fontos megemlíteni még a szociolingvisztikai kiegyensúlyozatlanság kérdését is, ugyanis a fennmaradt szövegek a társadalmi elit és az értelmiség nyelvét reprezentálják, mivel régebbi korokban az írástudás csak egy szűk réteg kiváltsága volt.

3. A történeti szövegek feldolgozása

A történeti szövegek különböző forrásokból származhatnak, melyek különböző feldolgozást igényelnek: vannak kézzel írott szövegek a nyomtatás feltalálása előttről és utánról, és vannak korai nyomtatványok. A nyomtatás feltalálása előtti korból származó kézzel írott szövegek egy részének már készült nyomtatott átirata. Erre láthatunk egy példát az 1. ábrán, amelyen egymás mellett szerepel a Margit-legenda 89r oldalának eredeti, kézzel írott változata és az abból nyelvtörténészek által készített nyomtatott átirat (P. Balázs et al. 1990). Itt azt lehet látni, hogy az átirat készítői törekedtek az ortográfiai hűségre, ami egy régi szöveg nyelvészeti szempontú vizsgálatához, illetve további felhasználásához feltétlenül szükséges.

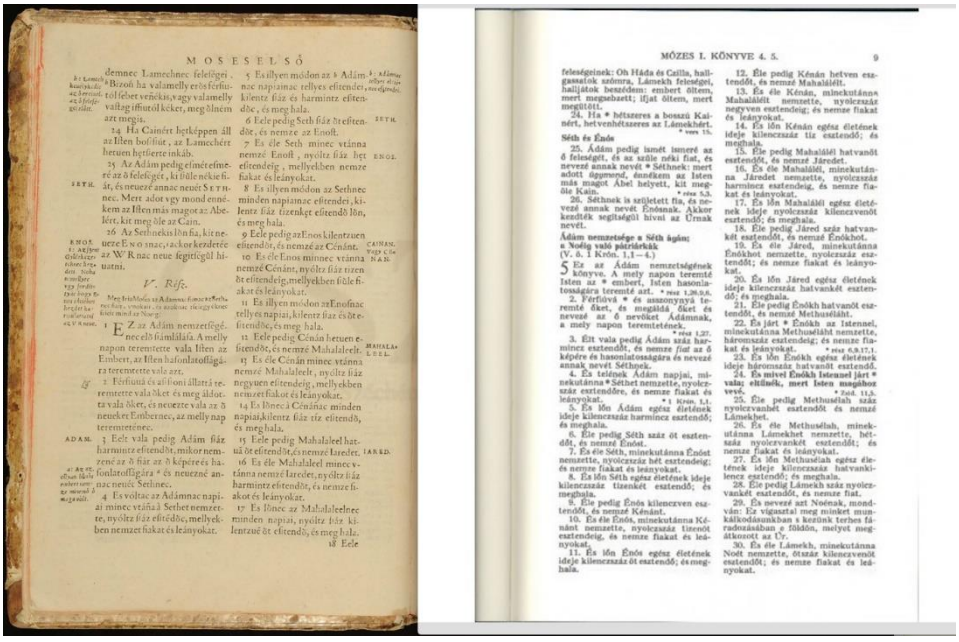
Ehhez hasonlóan léteznek a korai nyomtatványoknak későbbi, átdolgozott kiadásai is. A 2. ábrán a Károli Gáspár-féle bibliafordítás két kiadásának ugyanazon része szerepel. Bal oldalon az eredeti, 1590-es Vizsolyi Biblia (Károli 1590), jobb oldalon pedig az 1908-as revideált kiadás (Károli 1908) egy-egy oldala látható. Itt már jól tetten érhető, hogy az ún. revideált változat készítői nem fektettek nagy hangsúlyt a nyelvi jelenségek megőrzésére, hanem inkább a kor nyelvállapotához próbálták igazítani a régies nyelvezetet.



177
89r

tegedeth az kerestfan
sebhetted mereggel :
ees ecettel ytatal : ker-
-lek tegedet hog az
te zenth sebeyd legé-
-nek : en lelkemnek
orwossagy : amen
pr nr : Aue maria
O wram iesus
cristus ymadlak
tegedeth : coporsoba
helheztettel myrr-
-awal es zenth kenet-
-tel meg kenéttet

1. ábra: A Margit-legenda 89r oldalának eredeti, kézzel írott változata
és az abból készített nyomtatott átirat



MOSES EL SŐ

MÓZES I. KÖNYVE 4. 5. 9

2. ábra: A Károli-féle bibliafordítás egy-egy oldala két kiadásból.
Bal oldalon az 1590-es Vizsolyi Biblia, jobb oldalon az 1908-as revideált változat.

Amikor történeti korpuszt építünk, döntenünk kell, hogy milyen forrásokat használunk. A szerkesztett kiadások használata mellett szól egyrészt az, hogy általában könnyebben elérhetőek, mint az eredetiek. A Károli-féle bibliafordítás esetében, ha az interneten rákeresünk, akkor szinte az összes találat az 1908-as revideált változatra vagy annak valamelyik későbbi kiadására fog mutatni, míg az eredeti Vizsolyi Biblia szöveges változata csak az Ómagyar Korpusz weboldalán[3] érhető el. Szintén a szerkesztett kiadások használata mellett szól az, hogy a nyomtatott könyvek beszkennelése és azon optikai karakterfelismerő szoftver alkalmazása vagy a szöveg begépelése lényegesen könnyebb, mint ha az eredeti kézzel írott verzióval próbálná meg ugyanezt az ember. További fontos tényező, hogy a szerkesztői döntéseket már meghozták, így a történeti szövegek feldolgozásánál elég „csak" a feldolgozás lépéseire koncentrálni.

A szerkesztett kiadások használata ellen is szólnak azonban érvek. Amint már említettem, bizonyos kiadásokban az átírást és a szerkesztést nem nyelvészek végezték, hanem történészek vagy irodalmárok, akik egészen más szempontokat tartottak szem előtt, ám így is számos olyan döntést hoztak meg, amelyek hatással vannak a szövegre, de nincsenek kellően dokumentálva. Ezért ezek a kiadások nem alkalmasak további nyelvészeti vizsgálódásokra. Továbbá szerzői jogi problémákat is felvethetnek, hiszen az átirat készítői vagy élnek, vagy még nem telt el 70 év a haláluk óta, így az általuk szerkesztett kiadvány nem szabadon felhasználható, ellentétben az ómagyar kori kódexekkel, amelyek közkincsnek számítanak, nem köti őket szerzői jog. Kiutat jelenthet a dilemmából, ha szerkesztett kiadásokból indulunk ki, de mindent ellenőrzünk az eredeti verzióban.

A történeti szövegek feldolgozása és a belőlük való korpuszépítés számos kihívást rejt, amelyekkel a mai sztenderd nyelvállapotok feldolgozásakor nem feltétlenül szembesülünk. Az egyik ilyen a kézirat rossz fizikai állapotából következő nehézségek köre. A 3. ábrán a Königsbergi Töredék és Szalagjai elnevezésű korai szövegemlékünkből a Szalagok láthatóak. A nyelvemlék úgy maradt fenn, hogy egy másik, nem magyar nyelvű kódex kötéséhez használták fel: a számukra értelmetlen magyar szöveget tartalmazó lapokat kitépték, majd a Töredék lapját a kötet szennylapjául használták, a Szalagok lapját pedig a kötés megerősítésére csíkokra vágták, és beragasztották. Csak a 19. században fedezték fel a magyar nyelvű részeket, amikor kibontották, hazahozták, lefényképezték, de sajnos nem sikerültek túl jól a fotók. Viszont még a mai napig is csak ezekre a nem túl jól

[3] http://omagyarkorpusz.nytud.hu

sikerült fotókra kénytelen támaszkodni mindenki, aki ezt a nyelvemléket szeretné kutatni, ugyanis az eredeti Szalagok elvesztek (Madas 2009: 230–233).

3. ábra: A Königsbergi Töredék és Szalagjaiból a Szalagok

Az ómagyar kori szövegemlékeket és kódexeket a latin nyelvű és vallásos tárgyú irodalom fordításának igénye hívta életre, de a latin ábécé magyarra alkalmazása számos problémát vetett fel. A legfőbb gond abból fakadt, hogy nyelvünk hangrendszerének több eleme a latinban ismeretlen, így ezek jelölésére új jeleket kellett bevezetni. Kniezsa (1952) az ómagyar kori kódexek kezeinek helyesírását három nagy típusba sorolja. A mellékjel nélküli helyesírás a latinban nem szereplő magyar hangokat több betű kombinációjával írja le, például: *cs* → *ch* ~ *cz* ~ *chy* ~ *chi* ~ *cy*. A mellékjeles helyesírás egy rokonhang betűjének mellékjeles változatával jelöli ezeket, például: *cs* → *č* ~ *ć*. A harmadik típus pedig ezek keveréke, amely egy hang jelölésére karakterkombinációkat és diakritikus jeleket (akár egyszerre is) használ, például: *cs* → *ch* ~ *chy* ~ *cyh* ~ *c* ~ *chi* ~ *č* ~ *ch'*. Az ómagyar kor több mint 6 évszázadot fog át, amelynek során nem volt egységes hangjelölési rendszer, sőt egy kódexet akár több kéz is jegyezhetett, ami további egyenetlenségeket okoz a szövegekben. A különböző helyesírási rendszerekben is ritka az egy hang – egy betű megfelelés (vagyis amikor egy hang jelölésére mindig ugyanaz a betű használatos, és az adott betűnek mindig egy hangértéke van), de

egy alakulóban levő helyesírási rendszerben ilyenfajta következetesség még kevésbé van jelen. Sőt inkább az a tipikus, hogy egy emléken belül is ingadozik egy-egy hang jelölésmódja (pl. *kinec* [*kinek*]), vagy többes hangértéke van egy-egy betűnek (pl. *gimilcíctul* [*gyümölcsöktől*]). Tovább bonyolítja a helyzetet, hogy néhány betű egyaránt utalhat magánhangzóra és mássalhangzóra is, például az *u, v, w* több évszázadon át jelölhette az *u, ú, ü, ű, v, β* hangok bármelyikét (Korompay 2003). Ebből kifolyólag igen magas a speciális karakterek száma: az Ómagyar Korpuszban az 52 latin alapkarakter mellett 42 diakritikus jel, 10 szám, 34 szövegtagoló és egyéb jel, 3 görög betű, valamint 15 egyéb speciális karakter fordul elő, mindösszesen 156 karakter plusz ezek kombinációi. Vagyis már a betűhű szövegváltozat előállítása is sokkal nagyobb kihívást jelent, mint a mai magyar sztenderd nyelvváltozat esetében.

Ez a heterogén helyesírás az oka annak is, hogy a történeti szövegek esetében szükség van egy normalizáló lépésre, amelynek során az eredeti betűhű szóalakokat mai magyar helyesírású szavakra alakítjuk át. A normalizálás nehézsége abban rejlik, hogy egyszerre kell arra törekedni, hogy a helyesírási esetlegességeket kiküszöböljük, és eközben minden, ma már esetleg nem létező nyelvi jelenséget is megőrizzünk.

Ha magasabb szintű nyelvi annotációt szeretnénk a korpuszhoz adni, akkor ahhoz nyilván szükség van az adott szintű elemzőkre. Viszont az elérhető elemzők a modern nyelvállapotra készültek, ezeket alkalmassá kell tenni a régi nyelvállapot elemzésére, ami közelről sem triviális feladat. Akármilyen automatikus elemzőt használunk is, a kimenet mindenképpen kézi ellenőrzést igényel.

4. Magyar nyelvű történeti korpuszok

A magyar nyelvű történeti korpuszok sorában a magyar nyelv történeti szakaszait időrendi sorrendben követve az első az Ómagyar Korpusz. Ez a korpusz az MTA Nyelvtudományi Intézetében készül, a Magyar Generatív Történeti Szintaxis projekt keretében. A korpusz tartalmazza az összes fennmaradt ómagyar kori (896–1526) és néhány középmagyar kori (1526–1772) szövegemléket, valamint számos középmagyar bibliafordítást. A feldolgozott anyag 47 ómagyar kódexet, 24 rövidebb ómagyar szövegemléket, 244 misszilist (elküldött levelet), valamint 5 középmagyar kori bibliafordítást foglal magában, jelenleg mindösszesen 3,2 millió szövegszót. A korpusz egy része normalizálva és morfológiailag elemezve is lett; a normalizált alkorpusz mérete 807.691 token, a morfológiailag elemzett

alkorpusz mérete 285.070 token. A korpusz weboldalán keresztül a teljes anyag betűhű szövege, valamint néhány nyelvemlék normalizált és morfológiailag elemzett változata elérhető, valamint kereshető a grafikus korpuszlekérdező felület segítségével. A korpusz felépítéséről és a korpuszépítés lépéseiről részletes leírást szolgáltat a weboldal, valamint Oravecz et al. (2009, 2010); Simon and Sass (2012); Simon et al. (2011); Simon (2014); Simon and Vincze (2016).

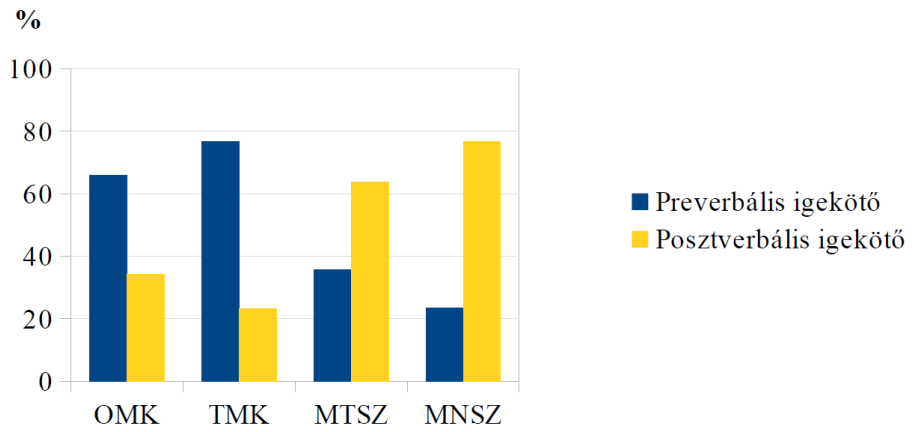
Az időben következő korpusz a Történeti Magánéleti Korpusz (Dömötör et al. 2017; Novák et al. 2018), amely szintén az MTA Nyelvtudományi Intézetében készült. A korpusz az ó- és a középmagyar kor magánéleti nyelvi regiszteréhez legközelebb álló műfajokat tartalmazza: magánlevelekből és peres eljárások jegyzőkönyveiből épül fel. Jelenleg körülbelül 850 ezer normalizált és morfológiailag elemzett szövegszót tartalmaz. A weboldalon[4] találunk leírást a felhasznált forrásokról, az alkalmazott morfoszintaktikai címkékről, valamint egy keresőfelületet és segédletet a használatához.

A Magyar Történeti Szövegtár (Csengery 2006) eredetileg a Magyar nyelv nagyszótárához (Ittzés 2011) készült, szintén az MTA Nyelvtudományi Intézetében. Ez a korpusz 1772-től, vagyis az újmagyar kor kezdetétől tartalmaz szövegeket a 20. század végéig. A régi keresőfelülete mellett készült hozzá egy újabb (Sass 2017), amely mögött a NoSketchEngine (Rychlý 2007) szabadon elérhető korpuszkezelő motor működik, és más magyar nyelvű korpuszok, mint például a Magyar Nemzeti Szövegtár (Oravecz et al. 2014) lekérdezőjéhez hasonlóan a CQL lekérdezőnyelvet használja. Ez a korpusz – a fenti kettővel ellentétben – nem tartalmaz semmilyen nyelvi annotációt, vagyis a keresésnél csak a felszíni szóalakra tudunk támaszkodni.

Mindhárom keresőre igaz, hogy elő tudunk velük állítani konkordancialistát és gyakorisági listát is. Mindhárom alkalmas arra, hogy történeti lexikológiai és szociolingvisztikai kutatásokhoz segítséget nyújtson. A morfológiai elemzést is tartalmazó korpuszok természetesen lehetőséget adnak történeti morfológiai vizsgálatok folytatására is, illetve bizonyos szintaktikai jelenségek is kutathatóak rajtuk. Például Kalivoda (2017) hat prototipikus igekötő (*meg*, *el*, *fel*, *ki*, *be*, *le*) szintaktikai viselkedését vizsgálta az ómagyar kortól napjainkig. A kutatás a felsorolt igekötők és a hozzájuk tartozó finit igék egymáshoz viszonyított helyét számolja, és a tagadó és a tiltó mondatokra jellemző igekötő–ige, illetve ige–igekötő sorrend arányát nézi. É. Kiss (2014) és Gugán (2015, 2017) azt állítja, hogy kétféle tagadó

[4] http://tmk.nytud.hu

szerkezet létezik a magyarban: a régebbi az egyenes szórendű (igekötő–tagadószó–ige, pl. *meg ne fogd*), az újabb pedig a fordított szórendű (tagadószó–ige–igekötő, pl. *ne fogd meg*). Állításuk szerint a fordított szórend is létezett a magyar nyelv korábbi szakaszaiban is, de csak a 19. századtól válik uralkodóvá. Kalivoda (2017) a preverbális (ige előtti) és a posztverbális (ige utáni) igekötők arányát vizsgálta a fent ismertetett három magyar történeti korpuszban, míg a mai nyelvállapot vizsgálatára a szintén említett Magyar Nemzeti Szövegtárt használta. A 4. ábra az egyenes és fordított szórendű tagadó mondatok százalékos arányát mutatja a vizsgált korpuszokban. A diagramon azt látjuk, hogy a posztverbális igekötőt tartalmazó tagadó mondatok arányának növekedése az ómagyar kortól napjainkig egyértelműen kimutatható a történeti korpuszok segítségével.



4. ábra: A preverbális és posztverbális igekötők arányaa Kalivoda (2017) által vizsgált korpuszokban

Irodalom

Biber, D. 1993. Representativeness of Corpus Design. *Literary and Linguistic Computing* 8/4.

Claridge, C. 2008. Historical corpora. In: Lüdeling, A., Kytö, M. (szerk.). *Corpus Linguistics. An International Handbook*. Berlin: Walter de Gruyter. 242–259.

Csengery K. 2006. Az elektronikus korpusz. In: Ittzés N. (szerk.). *A magyar nyelv nagyszótára 1. Segédletek.* Budapest: MTA Nyelvtudományi Intézet.

Dömötör A., Gugán K., Novák A., Varga M. 2017. Kiútkeresés a morfológiai labirintusból – korpuszépítés ó- és középmagyar kori magánéleti szövegekből. *Nyelvtudományi Közlemények* 113. 85–110.

É. Kiss K. 2014. A tagadó és a kérdő mondatok változásai. In: É. Kiss K. (szerk.). *Magyar generatív történeti mondattan*. Budapest: Akadémiai Kiadó. 34–49.

Gugán K. 2015. És mégis: mozog? Tagadás és igemódosítók az ómagyarban és a középmagyarban. *Általános Nyelvészeti Tanulmányok* 27. 153–178.

Gugán K. 2017. A magyar tagadó mondatok szórendje és a konstansrátahipotézis. In: *Nyelvelmélet és diakrónia 3*. Budapest; Piliscsaba: Pázmány Péter Katolikus Egyetem BTK; Szt. István Társulat. 91–110.

Hunston, S. 2008. Collection strategies and design decisions. In: Lüdeling, A. Kytö, M. (szerk.). *Corpus Linguistics. An International Handbook*. Berlin: Walter de Gruyter. 154–167.

Ittzés N. 2011. *A magyar nyelv nagyszótárának lexikográfiai koncepciója, különös tekintettel a szemantika és a grammatika összefüggésére a szótárírásban.* PhD-értekezés. Szeged: Szegedi Tudományegyetem.

Kalivoda Á. 2017. *Prototipikus igekötők mondatbeli helye az ómagyar kortól napjainkig*. Előadás a PPKE BTK Nyelvtudományi Doktori Iskolájának házi doktoranduszkonferenciáján.

Károli G. 1590. *SZENT Biblia, az az Istennec O es Wy testamentvmanac prophétác es apostoloc által meg iratott szent könyuei*. Vizsoly.

Károli G. 1908. *Szent Biblia, azaz: Istennek Ó és Új Testamentomában foglaltatott egész Szent Írás*. Magyar nyelvre fordította Károli Gáspár. Az eredetivel egybevetett és átdolgozott kiadás. Budapest: Brit és Külföldi Biblia-Társulat.

Kniezsa I. 1952. *Helyesírásunk története a könyvnyomtatás koráig*. Budapest: Akadémiai Kiadó.

Korompay K. 2003. Helyesírás-történet (az ómagyar korban). In: Kiss J., Pusztai F. (szerk.). *Magyar nyelvtörténet*. Budapest: Osiris Kiadó.

Madas E. (szerk.). 2009. *„Látjátok feleim...” Magyar nyelvemlékek a kezdetektől a 16. század elejéig*. Budapest: Országos Széchényi Könyvtár.

McEnery, T. 2004. Corpus Linguistics. In: Mitkov, R. (szerk.). *The Oxford Handbook of Computational Linguistics*. New York: Oxford University Press. 448–463.

Milić, L. 1990. The Century of Prose Corpus. *Literary and Linguistic Computing* 5/3. 203–208.

Novák A., Gugán K., Varga M., Dömötör A. 2018. Creation of an annotated corpus of Old and Middle Hungarian court records and private correspondence. *Language Resources and Evaluation* 52/1. 1–28.

Oravecz Cs., Sass B., Simon E. 2009. Gépi tanulási módszerek ómagyar kori szövegek normalizálására. In: Tanács A., Szauter D., Vincze V. (szerk.). *VI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2009)*. Szeged: Szegedi Tudományegyetem. 317–324.

Oravecz Cs., Sass B., Simon E. 2010. Semi-automatic Normalization of Old Hungarian Codices. In: *Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010)*. Lisbon: Faculty of Science, University of Lisbon. 55–60.

Oravecz Cs., Váradi T., Sass B. 2014. The Hungarian Gigaword Corpus. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC2014)*. Reykjavík: European Language Resources Association.

P. Balázs J., Dömötör A., Pólya K. (szerk.). 1990. *Szent Margit élete, 1510. A nyelvemlék hasonmása és betűhű átirata bevezetéssel és jegyzetekkel*, volume 10 of *Régi magyar kódexek*. Budapest: Magyar Nyelvtudományi Társaság.

Rissanen, M., Kytö, M., Kahlas-Tarkka, L., Kilpiö, M., Nevanlinna, S., Taavitsainen, I., Nevalainen, T., Raumolin-Brunberg, H. (szerk.). 1991. *The Helsinki Corpus of English Texts.* Helsinki: University of Helsinki.

Rychlý, P. 2007. Manatee/Bonito – A modular corpus manager. In: *Proceedings of the 1st Workshop on Recent Advances in Slavonic Natural Language Processing*. Brno: Masaryk University. 65–70.

Sass B. 2017. Keresés korpuszban: a kibővített Magyar történeti szövegtár új keresőfelülete. In: Forgács T., Németh M., and Sinkovics B. (szerk.). *A nyelvtörténeti kutatások újabb eredményei IX*. Szeged: SZTE Magyar Nyelvészeti Tanszék. 267–277.

Simon E. 2014. Corpus building from Old Hungarian codices. In: É. Kiss K., (szerk.). *The Evolution of Functional Left Perpiheries in Hungarian Syntax*. Oxford: Oxford University Press. 224–236.

Simon E. Sass B. 2012. Nyelvtechnológia és kulturális örökség, avagy korpuszépítés ómagyar kódexekből. In: Prószéky G., Váradi T. (szerk.). *Általános Nyelvészeti Tanulmányok XXIV*. *Nyelvtechnológiai kutatások*. Budapest: Akadémiai Kiadó. 243–264.

Simon E., Sass B., Mittelholcz I. 2011. Korpuszépítés ómagyar kódexekből. In: Tanács A. Vincze V. (szerk.). *VIII. Magyar Számítógépes Nyelvészeti Konferencia*. Szeged: Szegedi Tudományegyetem. 81–89.

Simon E., Vincze V. 2016. Universal Morphology for Old Hungarian. In: *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Berlin: Association for Computational Linguistics. 118–127.

Bevezetés az `e-magyar` programcsomag használatába

Mittelholcz Iván
fejlesztőmérnök
MTA Nyelvtudományi Intézet
mittelholcz.ivan@nytud.mta.hu

2006 óta dolgozom az MTA Nyelvtudományi Intézetében szoftverfejlesztőként. Érdeklődési körömbe olyan nyelvtechnológiai feladatok tartoznak, mint a természetes nyelvi szövegek tokenizálása, a helyesírás-ellenőrzés vagy az ontológiaépítés. Foglalkozom felügyelt gépi tanulással, webprogramozással és projektek számára kényelmesen használható számítógépes infrastruktúra kialakításával is.

1. Mire jó?

A számítógépes nyelvészet természetes nyelvi szövegek automatikus elemzésével vagy gépi feldolgozásával foglalkozik, ezt a célt szolgáló szoftvereket fejleszt. A diszciplína története során kialakult a szövegfeldolgozási lépéseknek egy többé-kevésbé egymásra épülő lánca. Ez olyan lépéseket foglal magában, mint a szöveg felbontása mondatokra és szavakra, a szavak morfológiai elemzése, a szófaj megállapítása, a mondatok szintaktikai elemzése, továbbá bizonyos sajátos mondatalkotó kifejezések felismerése a szövegben (pl. tulajdonnevek).

Az `e-magyar` programcsomag is egy ilyen, a fenti elemzési pontokat megvalósító eszközlánc. A lánc egymásra utalt modulokból áll. Egy modul a láncban előtte lévő modul kimenetén kezd dolgozni, ahhoz hozzáteszi a maga elemzéseit, majd továbbadja az egészet a következő modulnak. Egy ilyen láncba az első modul a még teljesen elemzetlen szövegen kezd el dolgozni, míg az utolsó kimenete a már teljesen elemzett szöveg, amit már nem fog további modul feldolgozni. Az informatikában pipeline-nak, azaz csővezetéknek szokták ezt az architektúrát nevezni. Ahogy a fizikai csővezetékben – gondoljunk itt, mondjuk, a kőolajfinomításra – halad a kezdetben nyers olaj, és válik a feldolgozás során lépésről lépésre finomabbá[5], úgy halad át a szöveg is egy ilyen elemzőláncon, és válik fokozatosan egyre feldolgozottabbá.

[5] Hogy finom nem lesz, azt már Besenyő Istvántól is tudhatjuk.

A szöveg feldolgozottsága az úgynevezett annotációkban vagy címkékben ölt testet. Az egyes elemzőmodulok az általuk megállapított információkat ilyen címkékben fűzik hozzá a szöveghez. Az annotációnak vagy címkézésnek a formátuma meglehetősen sokféle lehet. Például egy XML-alapú jelölésben így nézhet ki egy szófaji címke: `<szó szófaj="főnév">alma</szó>`.

Az `e-magyar` elemzőlánc a főbb magyarországi nyelvtechnológiai műhelyek összefogásával készült.[6] A modulok többnyire nem a nulláról indultak, hanem a műhelyek már meglévő programjai lettek átdolgozva, összehangolva. Magyar nyelvre eddig egy hasonlóan komplett elemzőlánc készült csak, mégpedig a szegedi magyarlánc.[7] A magyarlánc egyes komponensei az `e-magyar`nak is részei.

Az `e-magyar` programcsomag a kezdetektől úgy lett tervezve, hogy mind a szakmabeli nyelvtechnológusok és fejlesztők, mind az érdeklődő nagyközönség hasznát tudja venni. A szakma számára az `e-magyar` nyílt forráskódú projektként elérhető és Linux operációs rendszerekre telepíthető a projekt GitHub oldaláról.[8] A nagyközönség számára az `e-magyar` honlapján keresztül elérhető az elemzőlánc egy online kipróbálható változata. A következőkben ezt mutatjuk be.

2. Hogyan működik?

Az online változat a http://e-magyar.hu/hu/parser címen érhető el. Bárki számára ingyen használható, viszont az egyszerre elemezhető szövegek terjedelme 6000 karakterben van korlátozva. Akinek ennél nagyobb igényei vannak, az kénytelen telepíteni a programcsomagot és a saját számítógépén elvégezni az elemzést.

A nyelvtechnológiában a feladatok megoldásának két alapvető módszere van. Az első csoportba a szabályalapú megközelítések tartoznak. Ezek közös jellemzője, hogy `ha …, akkor …` típusú szabályokból álló feltételrendszerekkel kezelik a problémákat.

A második csoportba az úgynevezett statisztikai megközelítések tartoznak. Ezekre az igaz, hogy nincsenek általános, kőbe vésett szabályok, hanem sok, már

[6] MTA Nyelvtudományi Intézet, Pázmány Péter Katolikus Egyetem, Szegedi Tudományegyetem, MTA SZTAKI, AITIA International Zrt., Morphologic Kft.
[7] https://rgai.inf.u-szeged.hu/node/100
[8] https://github.com/dlt-rilmta/e-magyar-tsv. Megjegyzendő, hogy a projekt most egy nagyobb átdolgozáson esik át. Ez az új változat URL-e. A régebbi változat elérhetősége: https://github.com/dlt-rilmta/hunlp-GATE.

megvizsgált adat alapján állít fel az elemző egy statisztikai modellt, és az elemzés során ez a modell mondja meg, hogy milyen címkével kell ellátni az elemzendő nyelvi kifejezést.

Az `e-magyar` programcsomag egyes elemzői szabályalapúak, míg mások statisztikaiak. Hogy mikor és melyik módszert érdemes használni, az a problémán, az elemzési feladaton múlik. A szabályalapú módszer gyors és könnyen tud kezelni egyszerűbb, jól körülhatárolható feladatokat, de könnyen válik emberileg átláthatatlanná, nehezen javíthatóvá, ha egyszerre sok feltételt, alesetet kell kezelni. Megszokott tapasztalat az elbonyolódó szabályrendszereknél, hogy egy dolog kijavításával több új hibát „vezet be" az ember.

Ilyen esetekben lehet hasznos a statisztikai megközelítés,[9] ami jól kezel sok, akár logikátlan, kivételeket tartalmazó esetet is egyszerre. Azonban ennek is megvan a maga hátránya: sokszor a programok fejlesztői vagy használói maguk sem tudják, hogy miért úgy működik az elemzőjük úgy, ahogy.

A most következő alfejezetekben röviden azt ismertetjük, hogy az elemző egyes moduljai mit csinálnak, és azt hogyan teszik.

2.1. Mondatra bontás, tokenizálás

Az első modul két dolgot is csinál egyszerre: a nyers szöveget először mondatokra bontja, aztán a mondatokat úgynevezett tokenekre.

A mondatra bontásnak kettős szerepe van. Egyrészt vannak olyan elemzők a láncban, amelyek mondatokon dolgoznak, nem csak szavakon (ilyenek például a szintaktikai elemzők), másrészt a szöveg szavakra bontása is könnyebbé válik, ha már megvannak a mondathatárok. A mondatra bontás egyszerű feladatnak tűnhet (és igazából az is), de van pár nehézség benne. Ilyen például a rövidítések megfelelő kezelése. A következő példamondatban például kifejezetten hiba a pont és nagybetű között mondathatárt sejteni.

Támogatta a haladó eszméket, barátságban állt pl. Jókai Mórral is.

A tokenizálás feladata a mondathatárok között megkeresni az értelmes, a morfológiai elemzőnek továbbadható szavakat. El kell különíteni a szóközöket, az

[9] Ezt szokták gépi tanulás névvel is illetni, abból a hasonlatból kiindulva, hogy ha sok adatot mutatunk egy gépnek (igazából egy programnak), akkor az olyan, mintha „tanítanánk".

írásjeleket, megfelelően kell kezelni a dátumokat, mértékegységeket, az olyan informatikai kifejezéseket, mint e-mail-címek, URL-ek stb.

Mivel a mondatra bontás és a tokenizálás is viszonylag egyszerű feladatnak számít, ezért az `e-magyar`ba is szabályalapon működő program került.

2.2. Morfológiai elemzés

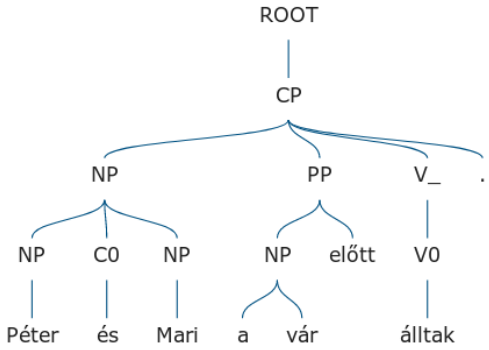
A következő elemzési szint a morfológiai elemzés. A tokenizálóból kijövő szavakat a morfológiai elemző egyenként beolvassa, és minden egyes szóhoz hozzárendeli az összes lehetséges morfológiai elemzését. A morfológiai elemzés tartalmazza a szóalak felbontását toldalékokra, képzőkre és összetételi határokra. Az elemző nem veszi figyelembe a szavak kontextusát, nem figyel arra, hogy az adott szó a mondat elején vagy végén található-e, hogy milyen más szavak vannak a környezetében stb. Csak magát a szóalakot veszi figyelembe, és próbálja azt minden lehetséges módon felbontani. A morfológiai elemzés ezért sokszor hoz meglepő eredményeket. Így lesz például a fenti példamondat *haladó* szavának öt lehetséges elemzése, köztük olyanok, mint a *hal* + *adó,* azaz 'halakra kivetett adó', vagy a *hal* + *ad* + *ó,* azaz 'halakat adományozó'.

Az `e-magyar`ban lévő morfológiai elemző is szabályalapon működik, konkrétan egy véges állapotú transzdúcernek nevezett technológia van mögötte.

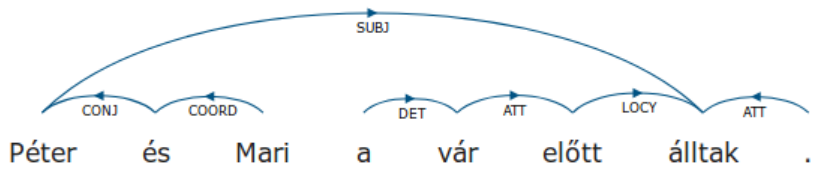
A morfológiai elemzőhöz szorosan kapcsolódik a szótövesítő modul. Ennek feladata, hogy a teljes morfológiai elemzésből előállítsa az elemzett szó szótári alakját és szófaját.

2.3. Morfológiai egyértelműsítés

A morfológiai elemzés kimenete tartalmazza minden szó minden lehetséges elemzését. Az egyértelműsítés feladata, hogy megállapítsa, a lehetséges elemzésekből melyik a helyes az adott mondatban. Például a „*Péter Marira vár*." mondatban a *vár* egyes szám harmadik személyű, jelen idejű ige, míg a „*Péter a vár előtt áll*." mondatban a *vár* alanyesetű főnév.



1. ábra: Összetevős elemzés



2. ábra: Függőségi elemzés

Ahhoz, hogy a sok lehetséges elemzés közül ki lehessen választani a megfelelőt, már nem elég a szavakat magukban nézni. Az egyértelműsítő modul már mondatokat vizsgál, és a lehetséges elemzések legvalószínűbb láncát keresi.

Az egyértelműsítés már nem szabályalapú, hanem statisztikai módszereken nyugszik. A módszer lényege, hogy sok, már elemzett és emberek által egyértelműsített mondatot mutatunk a programnak. Az egyértelműsítő ez alapján megtanulja, melyek a tipikus vagy gyakoribb mintázatok egy szövegben, és melyek a valószínűtlenek. Nem egyszerűen azt nézi, hogy a *vár* gyakrabban ige, mint főnév. Számára az a fontos, hogy egy névelőt sokkal gyakrabban követ főnév, mint ige, ezért az „*a vár*” kifejezésben a *vár*-at főnévként egyértelműsíti. Hasonlóan, a *Marira vár* esetében a szublatívuszi esetragú főnév után valószínűbb az ige, mint a főnév, ezért az igei címkét fogja kapni az elemzőtől.

Ehhez hasonló, feltételes valószínűségeken alapuló számításokkal tudja a program megállapítani a legvalószínűbb címkesorozatot minden mondatnál.

2.4. Szintaktikai elemzés

A szintaktikai elemzés, az egyértelműsítéshez hasonlóan, szintén mondatokon működik, és morfológiailag már egyértelműsített szöveg mondatainak építi fel a szintaktikai fáját. Az `e-magyar` programcsomag kétféle szintaktikai elemzőt tartalmaz.

A Chomsky-féle generatív nyelvtanon alapuló összetevős elemzés a mondatot kisebb kifejezésekre bontja (főnévi, igei stb. frázisokra), egészen addig, amíg el nem jut a mondatot alkotó szavakig. A fa leveleit a mondat szavai (terminálisok) alkotják, a fa nem-levél csomópontjai pedig az egyre összetettebb kifejezések (nem-terminálisok). A fában lévő élek címkézetlenek (lásd az 1. ábrát).

A függőségi elemzés olyan elemzési fát ad kimenetül, ami a szavak közötti relációt fejezi ki. A fa minden csomópontja egy szó, minden csomópontok közötti él a gyerek (függő) csomóponttól mutat a szülő csomópont felé. A függőségi fa élei a függőségi viszony típusával vannak címkézve (lásd a 2. ábrát).

Mindkét szintaktikai elemző statisztikai módszerrel dolgozik, és a már említett magyarláncból lett átvéve. Az összetevős elemző a Berkeley parser,[10] a függőségi elemző pedig a Bohnet parser[11] magyarra igazított változata.

2.5. Főnévi csoportok és tulajdonnevek felismerése

A következő két elemző nagyon hasonló elven működik. Az egyik maximális főnévi kifejezéseket keres a szövegben,[12] a másik a tulajdonneveket igyekszik felismerni. Mind a két feladat az információkinyerésnek nevezett problémához jelent segítséget. Az információkinyerés célja, hogy természetes nyelvi szövegekből olyan alapvető információkat nyerjen ki, mint hogy ki, kivel, mikor és mit csinált. Ezek az információk alapvetően NP-k vagy tulajdonnevek által jelölt entitások közötti relációkat írnak le. Ahhoz, hogy az entitások közötti relációkat sikerüljön egy szövegből automatikusan kinyerni, előbb fel kell ismerni magukat a szövegben szereplő entitásokat. Ezt a célt szolgálja ez a két modul.

[10] https://github.com/slavpetrov/berkeleyparser
[11] https://code.google.com/archive/p/mate-tools
[12] Maximális főnévi kifejezéseknek vagy NP-knek azokat az NP-ket nevezzük, melyek nem részei egy nagyobb főnévi kifejezésnek sem.

Mindkettő ugyanazon a gépi tanulási módszeren alapul.[13] Ennek lényege a szokásos gépi tanulásos eljáráson alapul: sok olyan szöveget mutattunk a programnak, amiben már be vannak jelölve a tulajdonnevek vagy a maximális NP-k, és megmondjuk azt is a programnak, hogy a szövegben a szavak milyen egyéb tulajdonságaira figyeljen. Például: hogy kis- vagy nagybetűvel kezdődik-e, hogy mondat elején van-e, hogy milyen a szófaja stb. A program megpróbált összefüggéseket találni ezen tulajdonságok és a címkék között, és előállított egy ún. valószínűségi modellt. Az `e-magyar`ba beépítettük ezt a valószínűségi modellt, amit az elemzőmodul arra használ, hogy a címkézetlen szövegben a szavak tulajdonságai alapján megpróbálja a címkéket megállapítani.

3. Mik a korlátai?

A nyelvtechnológiai rendszerek nem tökéletesek, ez sok feladat esetében elvileg is lehetetlen. A cél általában nem is egy tökéletes rendszer elkészítése, hanem egy „elég jó” rendszeré.

A hasonló feladatokra készült elemzőket két fő paraméter alapján szoktuk összehasonlítani. Az elsőt pontosságnak (precision) nevezik, és ez annak arányát jelenti, hogy a kiosztott címkékből mennyi a helyes. Például, ha az elemzendő szövegben száz kifejezést jelölt tulajdonnévként a program, és ebből nyolcvan volt helyes, akkor a rendszerünk pontossága 0,8. A másik mérőszámot fedésnek (recall) nevezzük, és ez az eltalált és a szövegben ténylegesen meglévő dolgok arányát jelenti. Például, ha a szövegben száz maximális főnévi kifejezés volt, és a rendszerünk ebből hetvenet talált el, akkor a fedés 0,7 lesz.

Az elemzők ilyenfajta kiértékelése feltételezi, hogy van olyan szövegünk, amelyet emberi munkával is elemeztünk, és ezt az elemzést biztosnak fogadjuk el, hogy összehasonlíthassuk a gépi elemző kimenetével.

4. Ajánlott irodalom

Az `e-magyar` eszközláncról négy cikk is megjelent a 2017-es Magyar Számítógépes Nyelvészeti Konferencia kiadványában.[14]

[13] Mindkettő a HunTag3 programot használja, lásd https://github.com/ppke-nlpg/HunTag3

[14] Elérhető a https://rgai.inf.u-szeged.hu/sites/rgai.sed.hu/files/kotet.pdf címen.

1. Általános áttekintést nyújt az `e-magyar` programcsomag egészéről:

Váradi T., Simon E., Sass B., Gerőcs M., Mittelholcz I., Novák A., Indig B., Prószéky G., Farkas R., Vincze. V. 2017. Az `e-magyar` digitális nyelvfeldolgozó rendszer. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 49–60.

2. Egy cikk az `e-magyar` tokenizáló moduljáról:

Mittelholcz I. 2017. emToken: Unicode-képes tokenizáló magyar nyelvre. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 61–69.

3. A morfológiai elemző alapos bemutatása:

Novák A., Rebrus P., Ludányi Zs. 2017. Az emMorf morfológiai elemző annotációs formalizmusa. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 70–78.

4. Az `e-magyar` GATE-integrációjáról szól, de a bennefoglaltak már nem feltétlenül helytállók. Az `e-magyar` jelentős átdolgozáson esett át mostanában, aminek egyik fontos része pont a GATE keretrendszertől való megszabadulás volt:

Sass B., Miháltz M., Kundráth P. Az e-magyar rendszer GATE környezetbe integrált magyar szövegfeldolgozó eszközlánca. In: Vincze V. (szerk.). *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. *MSZNY 2017*. Szeged: Szegedi Tudományegyetem, Informatikai Intézet. 79–90.

A kötetben található további két cikk, melyek szintén az `e-magyar` projekt keretében készült eszközökről szólnak, de ezek nem képezik szerves részét az ismertetett eszközláncnak.

A magyar nyelv és nyelvtechnológia helyzetét ismerteti:

Simon E., Lendvai P., Németh G., Olaszy G., Vicsi K. 2012. *A magyar nyelv a digitális korban. The Hungarian Language in the Digital Age*. Springer.

A http://www.meta-net.eu/whitepapers/e-book/hungarian.pdf címen letölthető könyv külön figyelmet fordít arra, hogy a jelen kor kihívásainak fényében tárgyalja a nyelvészeti és nyelvtechnológiai témákat.

Gyakorlatorientált bevezetést nyújt Mittelholcz Iván és Simon Eszter közös számítógépes nyelvészeti kurzusának anyaga. Ez a https://github.com/m-ivan/compling címen érhető el.

Ennél alaposabb és elméletibb bevezetést ad a nyelvtechnológia problémáiba és módszereibe:

Jurafsky, D., Martin, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River: Prentice Hall.

A klasszikus könyv készülő, harmadik kiadása online elérhető a https://web.stanford.edu/~jurafsky/slp3 címen.

Kvalitatív és kvantitatív szövegelemzés szoftverrel

Juhász Valéria
főiskolai docens
SZTE JGYPK Magyar és Alkalmazott Nyelvészeti Tanszék
juhaszvaleria@jgypk.szte.hu

Kutatási területeim az anyanyelv-pedagógia, az olvasás tanítása és tanulása, az olvasási készség fejlesztése, valamint a médiában és számítógépeken zajló kommunikáció tartalomelemzése. A közelmúltban a MAXQDA tartalomelemző szoftver használati lehetőségeivel és használatának népszerűsítésével is foglalkoztam.

1. A szövegelemző szoftverekről[15]

A szövegelemző szoftverek ősei a szószámláló számítógépes programok voltak. Kezdetben a számítógépet csak információ-visszakeresési rutinműveletekre használták a szövegek elemzésekor. A szövegelemző szoftver[16] szókapcsolat azt sugallhatja, hogy léteznek olyan programok, melyek segítségével automatikusan, igen gyorsan hozzájuthatunk szövegekből származó egyéni vagy társadalmi jelenségeket / folyamatokat jelző, tükröző vagy azt meghatározó, illetve magyarázó fogalmakhoz, viszonyokhoz, jelentésekhez. Ezzel szemben a programok olyan eszközök, melyek segítséget nyújtanak ugyan a kutatónak a rendszeres, rendezett munkához, de a jelenségek magyarázata, a kódolás az elemzőre marad. A tartalomelemző szoftverek[17] a kódolás folyamatában nyújtják a legtöbb segítséget.

[15] A cikk első megjelenése: Juhász Valéria. 2008. A MAXQDA szövegelemző program. In: Nádor O. (szerk.) *A magyar mint európai és világnyelv. XVIII. Magyar Alkalmazott Nyelvészeti Kongresszus előadásai*. Budapest: MANYE, Balassi Intézet.

[16] Angolul gyakran így található: CAQDAS = Computer Assisted Qualitive Data Analysis Software vagy Computer Assisted Content Analysis.

[17] A tartalomelemzés nem más, mint közlemények, szövegek meghatározott célú elemzése. Azokat az elemeket tárja fel nemcsak szövegekben, hanem képek, filmek közlés jellegű megnyilatkozásaiban, amelyek nincsenek nyíltan kimondva, esetleg az alkotóban sem tudatos közlésként jöttek létre, mégis a dekódolóban akár csak a sejtés szintjén is, de megjelenik a jelentésük. Ez a sorok közötti, sorok mögötti olvasás, értelmezés képessége, amely a kódolás módjában van elrejtve (Antal 1976). A tartalomelemzés módszerének a megismételhetőség kritériumait úgy kell biztosítania, hogy az egyes elemzési egységek kódolása és értelmezése a leírt metodika alapján ismét elvégezhető legyen. A módszeres és objektív eljárás azt jelenti, hogy a vizsgálat folyamán világos és egyértelműen megfogalmazott szabályok alapján dolgozunk. Előre meghatározzuk, milyen elemek, szimbólumok, szavak stb. kerüljenek az egyes kategóriákba. Minden tartalomelemzés két munkafázisra bontható. Az első fázist nevezik durván a kódolás fázisának. Ebben a fázisban a szöveget kódoljuk, tehát szimbólumait, szavait stb. előre megállapított kategóriákhoz

Kódoláson olyan karakterek, szavak, szövegegységek megjelölését értjük, amelyek valamilyen szempontból egy témakörbe tartoznak, van valami közös sajátosságuk, amiben hasonlítanak vagy eltérnek. A közös pont lehet a kód neve.

A forgalomban lévő szoftverek közül a magyar szakirodalomban viszonylag kevésre van utalás a tartalomelemzéssel végzett kutatásokban. Csakúgy, mint az adatokban és eljárásokban, a szoftverekben is alapvetően kétféle programcsomagot kínálnak a fejlesztők, vagy ezek kombinációit: az egyik a kvantitatív kutatásokhoz nyújt segítséget, a másik a kvalitatívhoz.[18] A legrégebbi, ám ma is használt program a szótáralapú The General Inquirer,[19] amit a CHILDES[20] követett. Ma ismertebb kvalitatív programcsomag például az ATLAS/ti, Code-A-Text, Computer Assisted Qualitative Data Analysis Software (CAQDAS) Networking Project, The Ethnograph v4.0, Kwalitan 4.0, NUD*IST, MAXQDA, QDA Miner, winMAX.[21] Hazánkban a kutatási beszámolókból úgy tűnik, hogy az ATLAS/ti és a NUD*IST programokat használják többen. Meg kell említenünk még a készülőben lévő magyar NooJ rendszert, amelyet integrált nyelvelemző környezetnek (INYEK) is neveznek, róla bővebben a következő honlapon olvashatunk: http://corpus.nytud.hu/nooj.

Jómagam a MAXQDA-val dolgozom,[22] mert a nemzetközi szakirodalomban olvasható, hogy számos területen sikeresen alkalmazzák szociológusok, politoló-

---

soroljuk. A második, magasabb fázisát az interpretáció szakaszának nevezhetjük. Az első szakaszban nyert mennyiségi eredmények értelmezésére, magyarázatára, a mélyebb rejtett összefüggések feltárására, kikövetkeztetésére kerül sor. Fontos tartalmi mutatóvá válhat valaminek a hiánya is.

[18] A szoftverekről bővebben: http://academic.csuohio.edu/kneuendorf/content/cpuca/ccap.htm

[19] A The General Inquirer Stone nevéhez fűződik (1966), új formájában az interneten is megtalálható: http://www.wjh.harvard.edu/~inquirer/. A program a Harvard IV-4 szótárt használva kódolja és osztályozza a szöveget olyan értékek megállapításával, mint az Osgood-féle háromdimenziójú szemantikai differenciálskála (a skála az egyes kijelentéseket három dimenzió mentén osztályozza: pozitív-negatív, erős-gyenge, aktív-passzív), kiválogatja az érzelemmel töltött szavakat, megállapítja a kognitív orientációt stb. A program elvégez olyan összegző statisztikákat, mint a szószámlálás vagy a szógyakoriság.

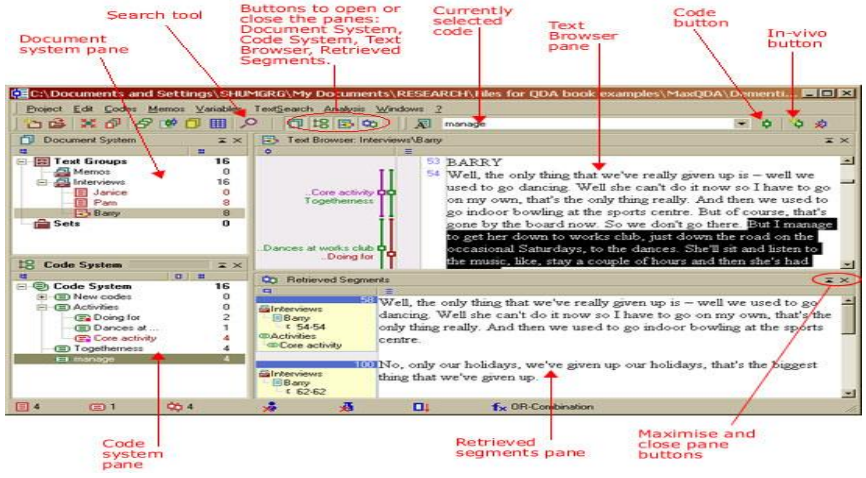
[20] CHILDES = Child Language Data Exchange. MacWhinney és Snow alkották meg (1985). Adatbázist, leíró rendszert és szerkesztő, kódoló programot, valamint egy morfológiai, szintaktikai elemző programot tartalmaz. Három (1,6 és 3,6 éves) iráni gyerek beszélgetéseinek elemzése.

[21] További szoftverek: http://www.qualitativeresearch.uga.edu/QualPage/qda.html

[22] A MAXQDA2 változat két fontos kiegészítést is tartalmaz. Az egyik a MAXDictio alprogram, amely szógyakoriságok és relatív gyakorisági mutatók összeállítására alkalmas, a másik pedig a Visio Tools, amelynek segítségével egyszerű ábrákat készíthetünk az eredmények szemléltetésére.

gusok, pszichológusok, egészségügyi kutatók, antropológusok, piackutatók, közgazdászok stb. Kiemelkedő tulajdonságai közül elsősorban a magas hatékonysági fokát, megbízhatóságát, stabilitását, jól kidolgozott funkcionalitását és felhasználóbarát felületét említik. A program egyszerűen kezelhető és világos struktúrával rendelkezik. A szoftver úgynevezett projektekkel dolgozik. Ezek a munkalapok tartalmazzák a szövegeinket, a kódokat, az emlékeztető megjegyzéseinket, a változóinkat, a kódolt szövegek összesített táblázatát, amelyeket attribútumokként (attributes) tárol a fájlon belül, és amelyekből statisztikai jellemzőkhöz is juthatunk. A különböző funkciók: a szövegben való keresés, a kódolás folyamata és a memóírás lehetősége azonos felületen jelennek meg. A projektet kezelő ablakban legfelül található a menüsor, alatta az eszköztár, amely a gyakran használatos – vagy a gyors hozzáférési lehetőségeket biztosító – gombokat tartalmazza. Alattuk a képernyőn négy ablakot láthatunk egyszerre: a bal fölső részen a teljes adatállományt (Document System/Text Groups), a bal alsó részen a kategória- vagy kódrendszert (Code System), a jobb fölső ablakban a kiválasztott, vizsgálandó / vizsgált szöveget magát (Text Browser), és végül a bal alsó sarokban látható az alapvető vagy komplex keresési eredményeket tartalmazó ablak. Ez a Retrieved Segments, amelyen nemcsak kialakítani lehet feltevéseinket, de igazolni is. Itt olvashatjuk az azonos kódolású szövegeket, illetve kódok együttes előfordulásából nyert szövegegységeket is, amelyekhez a program analitikai funkciójának segítségével juthatunk.

Az 1. ábrán látható a program felülete.



1. ábra: A MAXQDA felülete, forrás:
http://onlineqda.hud.ac.uk/Step_by_step_software/MAXqda/MAXqda_import_docs.php

2. A szövegek előkészítése, létrehozása és behívása a MAXQDA-ba

A Document System ablakban láthatók az elemzésre előkészített szövegeink, szövegcsoportjaink. Ebbe az ablakba az elemzés elkezdése után is illeszthetünk még újabb szövegeket. Bármilyen szöveget, szövegeket behívhatunk a programba, amely Rich Text Format (.rtf) kiterjesztésű dokumentum. A dokumentumban lehet ábra, Excel-táblázat, PowerPoint-elem, vagy tartalmazhat hyperlinket is, azonban Wordben vagy Office-ban szerkesztett nagyobb táblázatot nem tud kezelni a program, ezeket át kell alakítani szöveges formátumúvá.

Az .rtf kiterjesztésben elmentett szövegcsoportot többféleképpen lehet importálni a projektbe. Vagy úgy, hogy több szöveget egy egységként hívunk be, vagy úgy, hogy sok, viszonylag rövid, előre szerkesztett szöveget külön-külön egységként hív be a program. Szöveget azonban nemcsak importálni lehet a projektbe, hanem magában a projektben is létrehozhatjuk a munkaanyagunkat. Dokumentumainkat rendezhetjük ábécésorrendbe, de egyesével is mozgathatjuk őket, törölhetők, illetve új szövegek illeszthetők a rendszerbe. A Document System ablakocskán belül található egy Text Set nevű lehetőség. Ide úgy rendezhetjük a szövegeinket, hogy a meglévő kódok alapján újabb kis csoportokat hozunk létre. Kijelölhetjük neki az elemzett szövegnek egyszerre több tulajdonságát. Például, hogy csak azokat a szövegeket válogassa ki a program, amelyek húsvétkor készültek és nők írták (amennyiben ezek a tulajdonságok már kódoltak), és a Text Setbe illesztjük őket, hogy további megfigyeléseket végezzünk rajtuk.

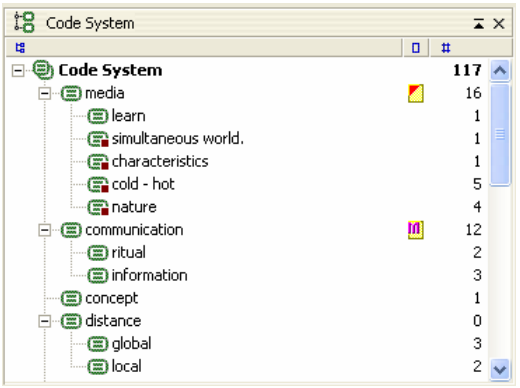
A szövegcsoportokon belül különböző lehetőségek vannak arra, hogy kijelöljünk, aktiváljunk szövegeket, mert azokkal akarunk dolgozni. Az Activate All gomb megnyomásával ki lehet jelölni az összes szöveget, a Ctrl gomb segítségével több szöveget is kijelölhetünk egyesével, valamint a szövegeink általunk megadott változói / attribútumai alapján egyszerűen aktiválhatunk több különböző, azonban egy vagy több változóban megegyező szöveget. (Az attribútumokra később részletesen kitérek.)

3. Kódolás

A MAXQDA egyik fő funkciója a kódolás maga: az, hogy a szöveg bizonyos részeihez, szavaihoz vagy akár csak egyetlen betűhöz kódokat, kategóriákat rendelünk. Ezek a kódok mutatnak rá a szövegben lévői tartalmi, lényegi minták jelenlétére. Ez a tulajdonképpeni kvalitatív elemzés első része. A kvalitatív adatok a kódolás folyamán úgy alakulnak kvantitatívvá, hogy az azonos kódok előfordu-

lásának számát, gyakoriságát őrzi a projekt. Ezek az előfordulások egy olyan mátrixba kerülnek, amelyeket aztán SPSS-be vagy Excelbe exportálhatunk. A végén a kvantitatív adatokat kvalitatív elemzésnek vetjük alá.

A kódok száma korlátlan lehet. Ezek hierarchikusan rendezhetők, és a kódok rendszere fastruktúraként jelenik meg a képernyőn. A kód-, illetve kategóriarendszer tízszintű mélységig, részletességig működik.



2. ábra: A kódrendszer fastruktúrája[23]

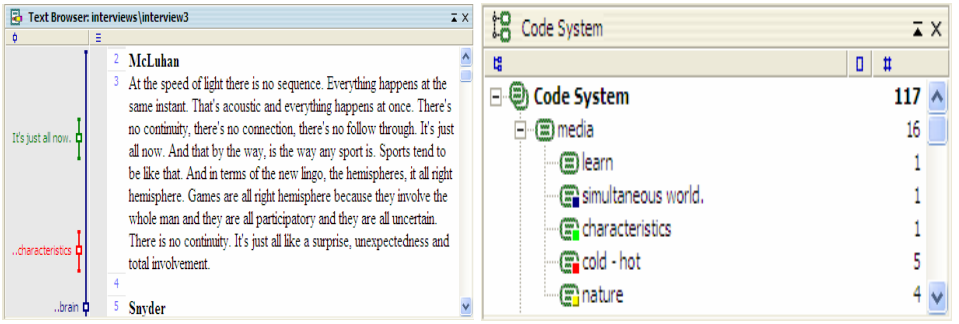
A kód létrehozása olyan, mintha egy üres fiókot alakítanánk ki. Ezeket „felcímkézve" belehelyezhetünk olyan dolgokat, amelyek a fiók elnevezésében egyeznek meg. Ha később úgy találjuk, hogy a fiók mégsem a legpontosabb nevet kapta, azt anélkül nevezhetjük át, hogy tartalmában bármiféle változás menne végbe.

A kódolás műveletét háromféleképpen hajthatjuk végre: automatikus rákereséssel (szavakra, szavak kombinációira, együttes előfordulásukra van lehetőség), illetve kézi kódolással kétféleképpen: az adatállományt vizsgálva kijelöljük a kódolandó elemet, majd ezt egy már előre kialakított kódhoz csatoljuk. A kódolás másik módja az in-vivo kódolás. Ez azt jelenti, hogy az adatállományban található elem lesz a kód neve is, azaz az elemet kijelölve, az in-vivo gombot megnyomva a kódot is létrehozzuk egyszerre. A program automatikusan legfölülre helyezi az in-vivo kódot, de a kutató ezt később egy áthelyezéssel könnyedén beilleszthetti a fastruktúrába. (A kódolásnál lehetőség van kódtörlésre is.)

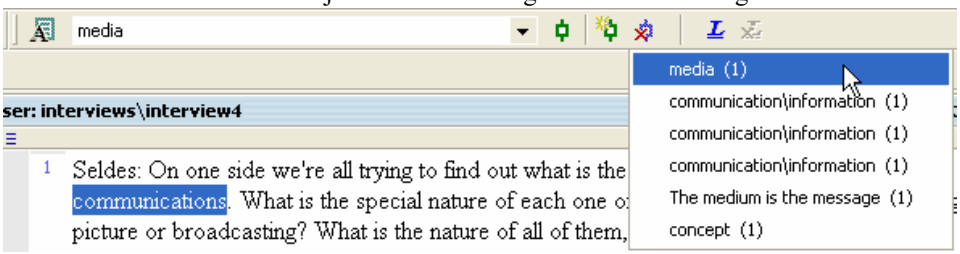
A kódok nemcsak a kategóriarendszerben jelennek meg, hanem az aktivált szöveg mellett is láthatóak, illetve átalakíthatóak attribútumokká, így megjelenik

[23] Az ábrákat a program útmutatójából vettem át.

majd a változók mátrixában. A programban arra is lehetőség van, hogy a kódjeleket különböző színekkel vizuálisan is elkülönítsük. Így olvasás, elemzés közben könnyebben azonosítjuk egy szöveg vagy szövegegység tartalmi vagy egyéb mutatóit. A kódolás során súlyozni is lehet a kódolt részeket aszerint, hogy azok mennyire relevánsak az adott esetben, azaz mennyire jellemzőek a szöveg és a kód összefüggésében.



3. ábra: Színes kód a kódolt szövegrész mellett és kódok a kódfában



4. ábra: A Text Browser ablakának eszköztárában látható kódolási lehetőségek

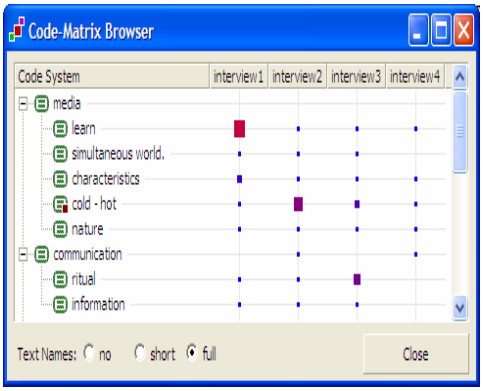
Kiemelkedő lehetősége a programnak, hogy egymást átfedő, egymásba ékelődő kódokat is képes kezelni. A kódfában a kódok nemcsak mozgathatók, hanem egymásba is másolhatók. Ezután viszont már nincs lehetőség az egymásba másolt, azaz egy kódhoz illesztett szövegrészek újabb szétválasztására automatikusan, csak manuálisan. (Ehhez kapcsolódóan megjegyzem, hogy a program egyik legnagyobb hátránya, hogy nincs rajta visszavonó gomb.)

A kódrendszert nemcsak kinyomtatni lehet, hanem egyik adatállományból a másikba exportálni, így – kódrendszerünket más adatállományon alkalmazva – hipotéziseinket igazolhatjuk, vagy újabb eredményekre juthatunk. A kódok előfordulásának gyakoriságát megtekinthetjük a Frequency of Codes alatt, illetve a

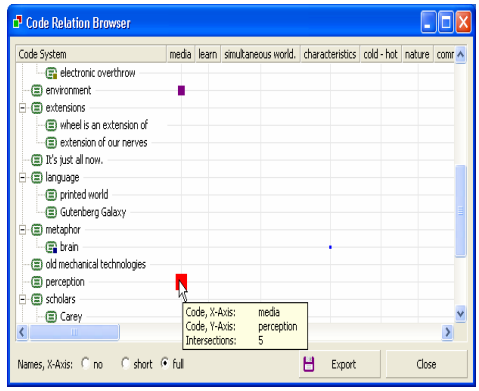
Code-Matrix Browser táblázatában. A Code Realation Browser a kódok közti kapcsolatot, összefüggést vizualizálja, azaz a szövegrészekhez rendelt kódok együttes előfordulását mutatja meg. Ezek a táblázatok segítenek kialakítani és igazolni is hipotéziseket, ezért jól alkalmazhatóak a Glaser és Strauss által leírt megalapozott elmélet (Glaser–Strauss 1967) technikai kivitelezésére.

	A	B	C	D	E	F	G
1	Textname	media	media\learn	media\simultaneous world.	media\characteristics	media\cold - hot	media\nature
2	interviews\interview1	0	11	3	4	3	3
3	interviews\interview2	0	1	2	1	8	2
4	interviews\interview3	0	1	3	2	5	2
5	interviews\interview4	0	1	0	1	2	1
6							

5. ábra: Kódgyakorisági-táblázat



6. ábra: Kódmátrix-böngésző

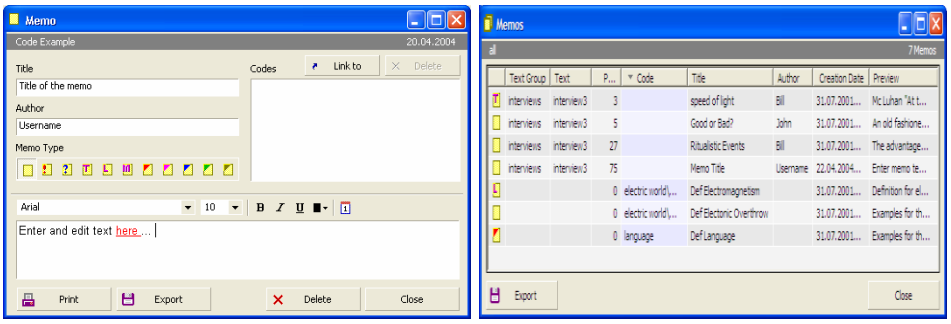


7. ábra: A kódok összefüggését mutató ábra

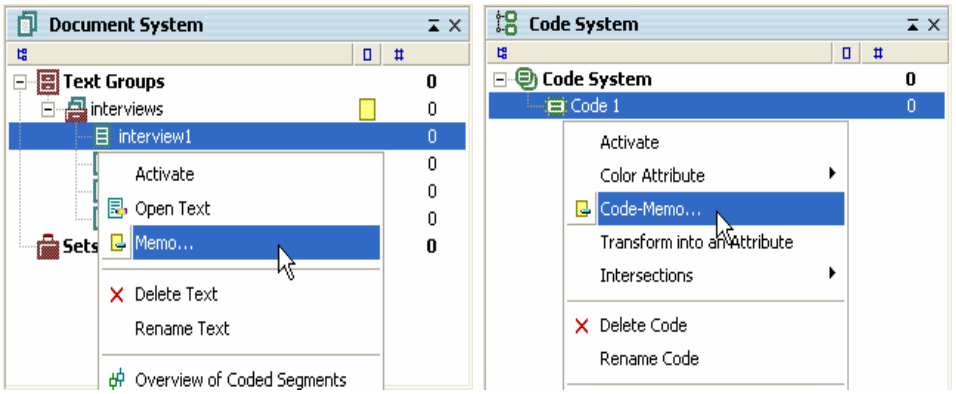
4. Emlékeztetők

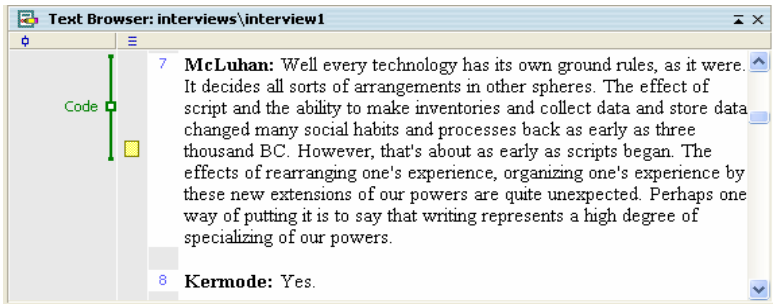
A hipotézis kialakításához rendelkezésre állnak az emlékeztetőket tartalmazó úgynevezett memók. Memók mind szöveghez, mind kódokhoz illeszthetők. A „cédulázásnak” ez a lehetősége biztosítja az egyedi esetek, illetve az aktuálisan megfigyelt, később kipróbálható összefüggések azonnali feljegyzését, valamint segít a kategóriarendszerről szóló fogalmaink végső pontosításánál is. A Memo Manager segítségével az összes feljegyzésünket egyszerre is megjeleníthetjük, illetve kereshetünk bennünk, ha elfelejtettük, hogy valamit hova írtunk fel. A memók között különböző típusú ikonok szolgálnak arra, hogy elkülönítsük például azt a megjegyzést, amely a kódra vonatkozik, vagy azt, ami a módszert vagy

akár az elméletet illeti. 10 ilyen különböző memo található a párbeszédablakban. A memókat ki lehet nyomtatni, exportálni lehet másik munkába; a program továbbá tartalmaz egy memorendszert, amelyben rögzítve van, hogy ki írta a feljegyzést, melyik szöveghez, illetve melyik kódhoz kapcsolódik, és itt olvasható a memo neve, a feljegyzés dátuma és a jegyzet első pár szava is. Ez csoportos munka esetében nagyon hasznos.



8. ábra: Memokészítő párbeszédablak és a memorendszer



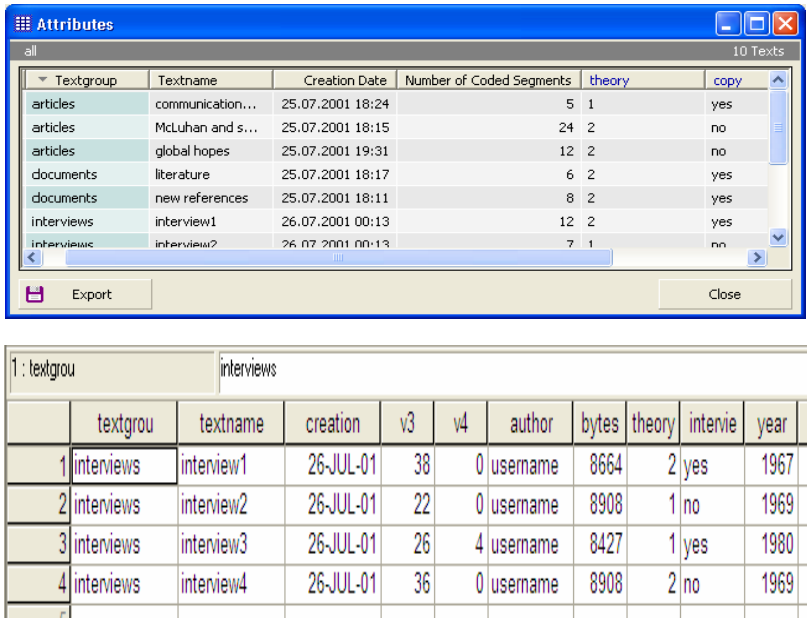


9. ábra: Szövegmemók, kódmemók és a szövegek melletti memók

5. Attribútumok

A MAXQDA-ban lehetőség van arra, hogy minden szöveghez jellemzőket, változókat rendeljünk. Változója lehet egy szövegnek a szövegalkotó neme, kora, lakóhelye stb., és a kialakított kódjaink is átalakíthatók ilyen attribútumokká, azaz változókká. Természetesen nemcsak szövegek lehetnek változók, hanem számok is. A változók megadásával kerestethetünk az adatbázisunkban. Például, ha meg akarjuk vizsgálni, hogy a 14 év és 20 év közötti nők milyen értékeket helyeznek előtérbe életükben mindennapjaikról szóló szövegeikben, akkor a változók segítségével csak a kritériumoknak megfelelő szövegek jelenítődnek meg (a náluk idősebbek vagy fiatalabbak szövegei nem).

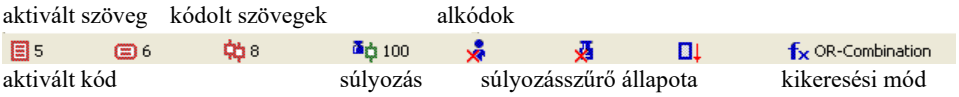
Ezeket a változókat a program egy négyszög alakú mátrixba rendezi. A mátrix mind a szövegek, mind a számok szerint rendezhető. Egy projektben csak egy ilyen attribútumokat tartalmazó táblázat van. A projekt kezdetén automatikusan megjelenik az attribútummátrixban a szövegcsoport neve, a szöveg neve, a létrehozás dátuma, a kódolt szakaszok száma, a memók száma, a szerző neve és a szöveg mérete byte-okban. Ebbe a táblázatba mi magunk is további elemeket illeszthetünk. Négy attribútumtípussal dolgozhatunk: szöveggel, számmal, dátummal és logikai változóval. (Ha később statisztikai programba akarjuk exportálni a mátrixot, akkor érdemes a numerikus jellemzőket használni.)



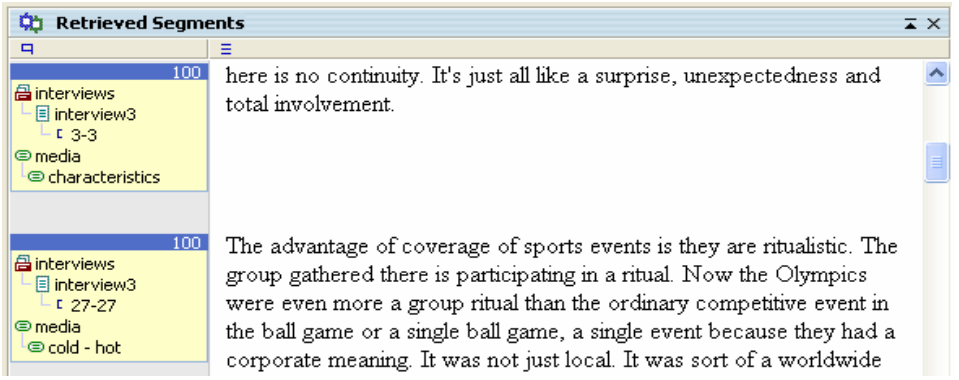
10. ábra: Attribútumtáblázat és SPSS-be illesztett megjelenése

6. Retrieved Coded Segments (RCS)

A képernyő bal alsó ablakában látható az ún. retrieved coded segments. A program itt jeleníti meg a már előzőleg kódolt és most kiválasztott / aktivált szegmenseket. Az újrakeresésnek két lépése van: először aktiválni kell a szövegeket, majd a kiválasztott kódot vagy kódokat. Ekkor a RCS-ben csak azok a szövegrészek jelennek meg, amelyeket előzőleg aktiváltunk. Az oldal alján lévő sorban látható a kikeresés összefoglalója: hány aktivált szövegünk van, hány aktivált kódunk, ebből hány újrakiválasztott elem lett a kikeresés eredménye.



11. ábra: Állapotjelző csík



12. ábra: A Retrieved Coded Segments ablaka

Hogyha rákattintunk a kikeresett szöveg összefoglaló ikonjára, akkor a Text Browser ablakban megjeleníti azt a szöveget, amelyben a kódolt szövegrész található, hogy további elemzéseknek vessük alá.

A szoftverben tíz analitikai funkcióval tudunk adatokat kinyerni. Az Analysis menü első opciója a logikai kombinációkat tartalmazza. Az aktivált szövegekben az aktivált kódokat előhívhatjuk a „vagy” kombinációval, egymásba illeszkedő, egymást átfedő kódokat kerestethetünk vele, vagy egymás közelében lévőket, illetve az „Only this code” azt jelenti, hogy csak a kijelölt kódokat tartalmazó részeket hívja elő, de a többit ne stb.

7. Zárógondolatok

A MAXQDA olyan csomag, ami lehetőséget nyújt a csapatmunkára is. Ez a tartalomelemzések egyik legfontosabb eleme, hiszen elősegíti és támogatja a kutatói elfogultsággal, prekoncepciókkal szembeni védekezést. A programba bevitt adatbázison különböző gépeken egyszerre lehet kódolni, majd a különböző fájlokat egymásba lehet illeszteni. A MAXQDA alkalmas arra, hogy – tartalomelemzéseik során – széleskörű lehetőségeivel segítse a kutatókat téziseik kialakításában.

A tartalomelemzés módszere egyre elterjedtebb a tudományos kutatások számos területén, mert jelentős eredményeket szolgáltat a társadalmi folyamatok megértéséhez, az összefüggések feltárásához. Bátran és sokrétűen alkalmazható mindenféle nyelvi anyag elemzésére: nyelvtanítási stratégiák javítására, nyelvfejlettségi szint meghatározására, ideológiák feltárására, szövegek összehasonlítására, különböző típusú szövegekben található azonos fogalmak meghatározására, a fogalmak politikai tartalmainak elkülönítésére stb.

Irodalom

Antal L. 1976. *A tartalomelemzés alapjai*. Budapest: Magvető.

Glaser, B. G., Strauss, A. L. 1967. *The Discovery of Grounded Theory; Strategies for Qualitative Research*. Chicago: Aldine Pub. Co.

Gyermeknyelvi korpuszok és erőforrások

Babarczy Anna
egyetemi docens
BME Kognitív Tudományi Tanszék; MTA Nyelvtudományi Intézet
babarczy@cogsci.bme.hu

Tanulmányaimat a University of Edinburgh-ban folytattam elméleti nyelvészetből, kognitív tudományból és mesterséges intelligenciából. Jelenleg a BME Kognitív Tudományi Tanszék tanszékvezető egyetemi docense és a Nyelvtudományi Intézet kutatója vagyok. Kutatási területeim a nyelvfejlődés és a pragmatikai kompetencia kísérletes vizsgálata és számítógépes modellálása.

1. Három gyermeknyelvi korpusz

Jelenleg, azaz 2019-ben, három viszonylag széles spektrumot átfogó, egészében magyar nyelvű – vagy magyar anyagot is tartalmazó – gyermeknyelvi korpuszt ismerünk, melyek: a GABI, a MONYEK és a CHILDES. A GABI (Gyermeknyelvi beszédAdatBázis és Információtár) fejlesztés alatt áll, és (egyelőre) nem elérhető a nagyközönség vagy akár a kutatóközösség számára, de egy rövid ismertető erejéig említést érdemel. A MONYEK (Magyar Óvodai Beszélt Nyelvi Korpusz) regisztrációval kutatási célokra elérhető a MetaShare szolgáltatáson keresztül az alábbi linken: http://metashare.nytud.hu/repository/browse/hungarian-kindergarten-language-corpus/b572a8106ba711e2aa7c68b599c26a06a4db2e695cf94a1cad6bf6793d747d2a/. A CHILDES (Child Language Data Exchange System) magyar és nem magyar nyelvű anyaga és eszköztára szabadon hozzáférhető, a korpusz bővíthető, tehát az alapszabályok betartása mellett bárki közzéteheti rajta keresztül a saját gyűjteményét. A CHILDES a TalkBank-rendszer részeként működik: https://childes.talkbank.org.

2. A CHILDES nemzetközi adatbázis és eszköztár

A CHILDES (Child Language Data Exchange System, Bernstein Ratner – MacWhinney 2016, 2018; Sagae et al. 2010) projekt 1984-ben indult Brian MacWhinney pszichológiaprofesszor szervezésében a Carnegie Mellon egyetemen. MacWhinney a doktori disszertációját óvodáskorú gyerekek morfológiai fejlődéséből írta, és ehhez gyűjtött spontánbeszéd-adatokat. A magyar nyelv gazdag morfológiája Magyarországra vonzotta, ahol öt óvodás magyar gyerekkel (Andi,

Éva, Gyuri, Móni, Zoli) készített felvételeket. Hazatérve az Egyesült Államokba, MacWhinney más kutatók gyűjteményeit is összeszedte, és a gyermeknyelvet kutató kollégái, valamint egy programozócsapat közreműködésével kidolgozott egy egységes átírási rendszert, és kifejlesztett számítógépes elemzőeszközöket a Unix operációs rendszerre. Mára a CHILDES adatbázisa 130+ különböző gyermeknyelvi korpuszt tartalmaz 30+ különböző nyelvből online, Windows-, Mac OS- és Unix-alapú elemzőprogramokkal együtt.

2.1. Az adatbázis

A CHILDES adatbázisában 2–9 éves, tipikusan fejlődő gyermekek spontán beszédén van a hangsúly, bár ennél fiatalabb és idősebb gyermekektől származó adatok is előfordulnak. Spontán beszéd alatt itt informális, kötetlen, a gyermek megszokott környezetében és megszokott családtagjaival zajló beszélgetést kell érteni. Ebben a kategóriában a korpuszok nagy része longitudinális, tehát nyomon követhetjük egy-egy gyermek nyelvi fejlődését éveken át, havi vagy akár heti rendszerességgel készült felvételeken keresztül. Az eredeti MacWhinney-féle és a később, mások által gyűjtött magyar adatok is ebbe a kategóriába tartoznak. A gyermekek többsége egynyelvű, de bilingvális korpuszok is találhatók az adatbázisban. A második legnagyobb kategória, ami a korpuszok összméretében messze elmarad a spontán beszédtől, a történetmesélés, azon belül is a Békamese elnevezésű, képekből álló történet elmeséléséről készült felvételek átirata. A Békamese- (angolul Frog Story) korpuszok Mercer Mayer amerikai gyerekkönyv író és illusztrátor 1969-ben megjelent munkáját, a Frog, where are you? (Béka, hol vagy?) című, 29, szavak nélküli, fekete-fehér vonalrajzból álló kalandtörténetét használják, amit a gyerekeknek a saját szavaikkal kell elmesélniük. A Békamese-korpuszoknak régi hagyománya van az angolszász gyermeknyelvkutatásban, ami már sok más régióra is kiterjedt. Jelenleg 13 nyelven találunk Békamese-korpuszokat a CHILDES adatbázisában. A morfológiai és szintaktikai fejlődés mellett a diskurzusjegyek és szövegkoherencia használatának vizsgálatára különösen alkalmasak ezek az adatok. A korpuszok harmadik kategóriája klinikai populációkkal készült felvételekből áll. Itt többek között Down-szindrómával, autizmussal vagy specifikus nyelvi zavarral élő, halláskárosult vagy agysérült gyerekekkel készült felvételek találhatók.

Az adatbázis a fenti kategorizáción túl nyelvenként (vagy nyelvcsaládonként), egy-egy nyelven belül pedig az adatgyűjtő kutató neve szerint rendeződik:

https://childes.talkbank.org/browser. A magyar adatokat az „Other” nyelvkategóriában találjuk a baszk, az észt és más magányos nyelvek társaságában. Három magyar korpusz található itt: az eredeti MacWhinney-korpusznak egy kibővített változata (MacWhinney 1974), Réger Zita longitudinális gyűjtése egy gyermekkel (Réger 1986), amely anyag a Nyelvtudományi Intézet jóvoltából még bővül, és Bodor Péter hasonló longitudinális korpusza egy kétéves gyermek első szavainak és mondatainak alakulásával (Bodor–Barcza 2007).

Az adatbázis elérhető online, de le is tölthető. A korpuszok három formában jelenhetnek meg az adatbázisban. A legegyszerűbb forma a hangfelvétel szöveges átirata – ez volt az eredeti, és sokáig egyetlen lehetséges mód, technikai korlátok miatt. A letölthető átirat a legegyszerűbb szövegfájlolvasóval is olvasható. Ma már lehetőség van az átirat és a hang összekapcsolására (a felhasználó hallja a hangot, és azzal párhuzamosan, a hangot valós időben követve látja az átírt szöveget), sőt az átirat, a hang és a mozgókép összekapcsolására is. Az adatbázis jelzi, hogy egyes korpuszok esetében elérhető-e hang-, illetve videófelvétel. A hanggal vagy videóval összekapcsolt átiratokat az online adatbázisban lehet követni, vagy letöltve a CHILDES szerkesztőszoftverével, a CLAN programmal lehet hallgatni, illetve szerkeszteni. A CLAN Windows, Mac és Unix operációs rendszerre is elérhető: http://dali.talkbank.org/clan. A későbbiekben visszatérünk még a használatára.

Bármely kutató kérheti a korpusza felvételét az adatbázisba, feltéve, hogy eleget tesz a CHILDES formai és etikai előírásainak. Ezek közül az egyik legfontosabb, hogy az adatgyűjtéshez etikai engedély szükséges, aminek a beszerzése történhet a kutatásvezető egyetemének vagy kutatóhelyének etikai szabályai szerint. A másik legfontosabb feltétel az, hogy a felvételeken szereplő gyerekek anonimizálva legyenek. Ez azt jelenti, hogy a felvételekből ki kell vágni minden olyan megnyilvánulást (vezetéknevet, címet, óvoda nevét stb.), ami a gyermeket azonosíthatja. Az etikai feltételek egyik következménye, hogy a videófelvételeket ritkán publikálják kutatók a nyilvános adatbázisban. A CHILDES természetesen a felhasználókat is felhívja a kutatóközösségben megszokott etikai normák betartására. A korpuszok szabadon hozzáférhetőek, de illik hivatkozni magára a CHILDES-ra és a felhasznált korpusz készítőjének megadott publikációira.

2.2. A CLAN program és az átírás szintaxisa

A hangfelvételek átirata és annotációi a CHAT formátumot követik. (Sajnos többszöri próbálkozással sem sikerült kiderítenem, hogy minek az akronimája vagy rövidítése lehet a CHAT betűsor.) Az átiratok .cha kiterjesztésű szövegfájlok a CHAT markereivel ellátva, amiket automatikusan konvertálni lehet más népszerű formátumokba, mint például Praat és ELAN. A szövegfájlokat bármilyen szövegszerkesztővel meg lehet nyitni, de az átíráshoz és annotációhoz érdemes a CHILDES saját szerkesztőprogramját, a CLAN szoftvert használni. A CLAN három fő üzemmódban működik: a chat mód az átírást könnyíti meg különböző funkciókkal, a sonic mód az átírás és hang/videó összehangolását, a coder mód pedig az annotációt. A CHAT és a CLAN részletes leírása megtalálható a neten: https://talkbank.org/manuals/CHAT.pdf.

A CHAT formátum a beszélt szövegek átírása mellett azok annotációjára is lehetőséget ad. A CHAT három típusú sort definiál: az egyik az átirat általános jellemzőit adja meg, és a `@` karakter vezeti be. A másik azt jelzi, hogy beszéd átiratát tartalmazza a sor. Ezt a `*` karakter és a beszélő hárombetűs kódja vezeti be. Egy beszélősorban csak egy mondat átirata szerepelhet, tehát minden mondat külön sorba kerül. Aki már próbált beszélt nyelvet átírni, nagyon jól tudja, hogy mennyire nem triviális kérdés a folyamatos beszéd mondatokra bontása. Praktikus okokból (az elemzés megvalósíthatósága miatt) azonban ez egy szükségszerű lépés. A harmadik sortípus a szöveg beszédsoronkénti annotációjára ad lehetőséget. Ezek a sorok egy-egy beszédsort követnek, és a `%` karakter és egy azonosító kód vezeti be őket. Az annotáció tartalmazhat morfológiai, szintaktikai vagy fonológiai elemzésen túl szociolingvisztikai markereket – vagy bármilyen egyéb megjegyzést, amit az átíró rögzíteni kíván. A sort bevezető azonosító kód jelzi, hogy milyen jellegű annotáció szerepel benne. Az (1) rövid részlet Réger Zita korpuszából jól illusztrálja a rendszert.

Az első két sor azonosítja a fájlt a CHILDES rendszerben. A `@Begin` kód jelzi, hogy kezdődik az átirat. A következő néhány `@` jelű sor megadja a nyelvet, a beszélők kódját és szerepét, a gyerek életkorát (2 év, 0 hónap, 25 nap), a hangfájl elérhetőségét, a dátumot, a hangfelvétel helyét a magnószalagon, az átíró nevét és a beszélgetés alaphelyzetét. Ezután következik a beszéd átirata, mondatonként. Ebben a részletben két annotációs sor van: a `%com` nevű megjegyzéseket tartalmaz arra vonatkozóan, hogy milyen helyzetbe került a gyerek, amikor éppen az adott

mondatot mondta, a %act nevű sor pedig a gyerek cselekedeteit írja le. Az átirat a @End kóddal végződik.

(1)

```
      @Loc: Other/Hungarian/Reger/020025.cha
1     @PID: 11312/c-00027754-1
2     @Begin
3     @Languages: hun
4     @Participants:    CHI Target_Child , MOM Mother
5     @ID:  hun|Reger|CHI|2;00.25||||Target_Child|||
6     @ID:  hun|Reger|MOM|||||Mother|||
7     @Media:     020025, audio, unlinked
8     @Date:      11-OCT-1992
9     @Tape Location:   Tape X , Side A. 212.
10    @Transcriber:     Szilvia Papp.
11    @Situation: Miki has just woken up. He is still in
bed.
13    *CHI: kivesz .
14    *CHI: anyu kivesz .
15    *CHI: anyu kivesz .
16    *CHI: anyu ki [//] anyu emej .
17    *MOM: jó kiveszlek , emellek .
18    *MOM: szervusz .
19    *MOM: jó reggelt !
20    *CHI: miki ajutt [=? aludt] .
21    %com: mother has lifted him out of the bed , miki is
sitting in her lap .
22    *MOM: hol aludt miki ?
23    *CHI: itt .
24    %act: points at bed .
...
31    *CHI: mag(n)ót .
32    *CHI: itt ?
33    %act: looking for the tape recorder .
...
903   @End
```

Angol (és néhány más) nyelvű szövegek morfológiai és szintaktikai elemzését automatikusan végzi a CHILDES. A morfológiai elemzés eredményét a %mor sorba, a szintaktikai elemzés eredményét pedig a %gra sorba illeszti. Magyar

nyelvre sajnos jelenleg egyik automatikus elemzés sem elérhető a CHILDES rendszerben, de lásd később a MONYEK korpusz leírását erre vonatkozóan. Egy angol példa a morfológiai és szintaktikai elemzésre a Manchester-korpuszból (Theakston et al. 2001):

(2)

```
@Situation: Structured Play
13    *CHI: I turned the cooker on .
14    %mor: pro:sub|I v|turn-PAST det:art|the n|cook&dv-AGT
prep|on .
15    %gra: 1|2|SUBJ 2|0|ROOT 3|4|DET 4|2|OBJ 5|2|JCT
6|2|PUNCT
16    *MOT: well done .
17    %mor: co|well part|do&PASTP .
18    %gra: 1|2|COM 2|0|ROOT 3|2|PUNCT
19    *MOT: cooking my pretzel ?
20    %mor: n:gerund|cook-PRESP det:poss|my n|pretzel ?
21    %gra: 1|0|INCROOT 2|3|MOD 3|1|OBJ 4|1|PUNCT
22    *CHI: it's cooked it already .
23    %mor: pro:per|it~aux|be&3S part|cook-PASTP pro:per|it
adv|already .
24    %gra: 1|3|SUBJ 2|3|AUX 3|0|ROOT 4|3|OBJ 5|3|JCT
6|3|PUNCT
```

A CHAT formátum szabályait és a megengedett annotációs (%) sorokat a depfile.cut nevű fájl tartalmazza, ami a CLAN programmal együtt telepítődik a számítógépre, ha a CLAN helyi, telepített változatát használjuk. Ez egy szövegfájl, ami a CLAN programmal (vagy bármely más szövegfájl szerkesztővel) szerkeszthető. Ez az, ami igazán rugalmassá teszi a rendszert, hiszen mindenki a maga igényei szerint definiálhatja az átirat szabályait. Az annotációs sorok kódjait is tetszés szerint definiálhatja a felhasználó további .cut fájlokban, ami lehetővé teszi például a magyar morfológiai elemzés beillesztését. Ezek a módosított, saját egyéni igények szerint kialakított kódrendszerek és .cut fájlok természetesen nem kerülnek be az online adatbázisba.

2.3. Az elemzőprogramok

A CLAN szerkesztő és az online adatbázis egy sor elemzőprogramot is kínál. A programokat parancssorral lehet behívni a CLAN szerkesztő egyik ablakában

vagy az online adatbázis bármelyik szintjén (a bal alsó sarokban). Az offline és online változat használata tökéletesen megegyezik egymással azzal az egy különbséggel, hogy míg az offline parancssorban specifikálni kell a mappát, amiben az elemzendő célfájlok vannak, addig az online parancssor automatikusan abban a mappában keresi a fájlokat, ahol éppen jár a felhasználó. A parancsok nem túl bonyolultak, a Unix operációs rendszer logikáját követik. A parancssor az elemzőalgoritmus nevével kezdődik, amit egy sor opció követ, majd az elemzendő fájlok neve zárja a parancsot. Ha azt szeretnénk megtudni például, hogy milyen b betűvel kezdődő szavakat milyen gyakorisággal használ a gyermek, a (3) parancsot adhatjuk:

(3) `freq -t% +t*CHI +s"b*" +u +o *.cha`

Itt a `freq` annak a parancsnak a neve, ami gyakorisági információt ad. A két `t` opció azt mondja, hogy ne az annotációs sorokban keresse a parancs a szavakat, hanem a gyermek beszédsorában (a sortípusokat tier-nek hívja a CHILDES, innen jön a `t`). Az `s` a keresendő string-et specifikálja: ebben az esetben a `b` betűvel kezdődő szavakat (egymástól nem-szó karakterrel elválasztott egységeket). Az `u` opció összegzi a fájlok keresésének eredményeit, az `o` opció gyakoriság szerinti sorrendben írja ki az eredményeket, a `*.cha` pedig a mappában található valamennyi .cha kiterjesztésű fájlt nevezi meg a keresés céljául. A parancs kimenetének első néhány sora MacWhinney Zoli-korpuszából a (4) alatt látható:

(4)

```
freq -t% +t*CHI +s"b*" +u +o *.cha
Thu Jan 24 12:09:19 2019
freq (27-Apr-2018) is conducting analyses on:
  ONLY speaker main tiers matching: *CHI;
****************************************
Speaker: *CHI:
474 bácsi
 46 bogár
 26 be
 23 bács
 18 bá
 11 bácsinak
 11 ba
```

```
 11 bemegyek
 11 bumm
  9 bírom
  9 baboda
```

Egy másik hasznos parancs a `kwal` (keyword and line), ami egy adott kulcsszónak a kontextusát mutatja meg. Ha például a fenti eredmény alapján azt szeretnénk megtudni, hogy miért mondogatta annyit a kétéves gyermek a *bírom* szót, az (5) szerint tehetjük meg:

(5) `kwal -t% +t*CHI +s"bírom" -w2 +w2 *.cha`

ahol a `-w` és `+w` azt specifikálja, hogy a célstring előtt és után két sort írjon ki a program. Az eredménynek egy részlete:

(6)

```
kwal -t% +t*CHI +s"bírom" -w2 +w2 +u *.cha
Thu Jan 24 12:25:41 2019
kwal (27-Apr-2018) is conducting analyses on:
  ONLY speaker main tiers matching: *CHI;
****************************************

From file "011000.cha"
----------------------------------------
*** File "011000.cha": line 401. Keywords: bírom, bírom
*CHI: jött (.) motor ott [% magának susog] ott teszem (.)
teszem (.) bujj el a róka kóma (.) itt van (.) bu bu (.)
itt is van (.) ott is van (.) kutyus .
*CHI: &=whisper (..) Barna bácsi , homokot [: homokba]
ülünk .
*CHI: együnk [//] egyetünk (.) itt itt [!] a nagy autó
(.) ott megyünk (.) alig bírom másik [% phrase] (.) &o
itt van a Vio [!] néni (.) autóval megyünk (.) alig
bírom .
*BRI: Zoli , tudod miért nem megy mert ki van szedve
belôle az elem (.) azért nem megy az autó .
*CHI: Barna bácsi (.) Barna bácsi .
```

```
From file "011001.cha"
----------------------------------------
*** File "011001.cha": line 552. Keyword: bírom
*CHI: fütyöl [: füstöl] .
*MON: egér (.) itt (.) egér (.) csüccsülj le , egér .
*CHI: alig bírom .
*BRI: ajaj , alig bír .
*CHI: Barna bácsi , dolgozunk .

From file "011002.cha"
----------------------------------------
*** File "011002.cha": line 544. Keywords: bírom, bírom,
bírom, bírom
*CHI: várj csak , Barna bácsi (.) várj csak .
@New Episode
*CHI: nagy (.) indulás (.) (.) alig bírom (.) elszakadok
(.) alig bírom (.) alig bírom (.) elszakadok (.) alig
bírom (.) elszakadok .
*UNK: hadd nézzem ezt a halacskát .
*CHI: jó itt (.) elromlott a +...
```

2.4. A CHILDES egyéb eszközei: a LuCiD Toolkit

A LuCiD Toolkit egy brit gyermeknyelv kutató társulás (az ESRC International Centre for Language and Communicative Development) eszköztára, melyet a CHILDES adatbázisának hatékonyabb kihasználására fejlesztett ki (Chang 2017). Az eszközök elérhetők a CHILDES oldalairól: http://gandalf.talkbank.org:8080. A CHILDES Browser lehetővé teszi a teljes adatbázis célirányos keresését bizonyos kritériumok szerint, mint például korpuszméret, a mondatok hossza vagy a gyermek kora. A Restricted Distribution eszköz feltérképezi azokat a szavakat, amelyek egy adott, reguláris kifejezéssel specifikált kontextusban előfordulnak. Megmutatja például, hogy a Bodor-korpuszban a mondatok legnagyobb valószínűséggel az *és, hát, nem, na, mit* vagy *igen* szóval kezdődnek. A CHILDES Generator megbecsüli egy adott kifejezést követő szavak bigram valószínűségét a korpusz alapján, a Distributional Word Classification eszköz pedig ngram valószínűségeket von ki a korpuszból és disztribúciós tulajdonságaik szerint rendezi a célszavakat.

3. A magyar gyermeknyelvi korpuszok

3.1. A MONYEK

A MONYEK (Magyar Óvodai Nyelvi Korpusz), Mátyus Kinga doktori disszertációjához készült hangfelvételekből és azok átiratából áll (Mátyus–Orosz 2014; Orosz–Mátyus 2014). A korpusz 62 óvodás gyerekkel folytatott interjú alapján készült; összesen mintegy 140 000 szót tesz ki. A gyerekek kiválasztásánál fontos szempont volt a társadalmi-gazdasági státuszuk. A gyerekek egyik felét a KSH adatai szerint elit budapesti óvodákból toborozták a kutatók, a másik felét pedig alacsonyabb társadalmi-gazdasági státuszú budapesti óvodákból. Az interjúk előre meghatározott forgatókönyv szerint zajlottak, az erősebben kontrollált feladatoktól a szabadabb társalgási feladatok felé haladva. Az első feladatban a kísérletvezető – képek alapján – elmondott egy történetet (Zsuzsi és az állatok), amit a gyermeknek vissza kellett mondania. Ezt követte egy önálló képmesélési feladat, ahol a fent már említett Békamesét és két rövidebb történetet meséltek el a gyerekek képek alapján. Ezután egy játék (pl. foci) menetét és szabályait írták le a gyerekek, és végül irányított beszélgetés zárta az interjút.

A hanganyag átírása a CHILDES CHAT formátumának szabályai szerint történt; a szöveg automatikus morfológiai elemzését a HuMor végezte, melynek kimenetét a PurePos egyértelműsítő rendszer gyereknyelvre adaptált változata egyértelműsítette, utólagos kézi ellenőrzéssel. A korpusz elérhető CHAT fájlok formájában és xml-formátumban. A MONYEK-korpusz készítését a CESAR-projekt és az MTA Nyelvtudományi Intézet támogatta.

3.2. A GABI

A GABI (Gyermeknyelvi beszédAdatBázis és Információtár) Bóna Judit irányításával készül az ELTE Fonetikai Tanszékén (Bóna 2017, Vakula–Váradi 2017). A korpusz különlegessége, hogy minden korosztályból gyűjt hangadatokat a 3 évesektől egészen a 18 évesekig, és így szinte minden nyelvfejlődési korszakot lefed. A korpusz értelemszerűen keresztmetszeti, tehát nem ugyanazokat a gyerekeket követi végig, hanem különböző gyerekektől gyűjt adatokat a különböző korosztályokban. A gyermekek kiválasztásának szempontja, hogy köznyelvet beszéljenek, és neurotipikus fejlődésűek legyenek. Részletes anamnézist vesznek fel velük, amelyben rákérdeznek a családi háttérre (szülők iskolázottsága, változás a családszerkezetben, anyagi helyzet, testvérek stb.), óvodára, iskolára, betegsé-

gekre, allergiára, nyelvfejlődési jellemzőkre (pl. járt-e a gyermek logopédushoz, mikor kezdett beszélni stb.).

Az adatfelvétel forgatókönyv szerint zajlik a BEA (BEszélt nyelvi Adatbázis, MTA Nyelvtudományi Intézet) elnevezésű felnőtt beszélt nyelvi korpusz módszertanát követve, és azt gyerekekre adaptálva. A feladatok egy része erősen kontrollált (mondatismétlés, mondatolvasás), más része valamivel több szabadságot ad (szavak definiálása, hallott történet tartalmának visszamondása), és végül kvázi-spontán beszédet kiváltó feladatokat is találunk (történetmondás képek alapján, szabad beszélgetés egy adott témáról). A forgatókönyv részletei a gyermek korának megfelelően kisebb mértékben változnak.

E tanulmány megírásáig több mint 450 gyermekkel készült felvétel, melyek hanganyagának átírása és annotálása folyamatban van. Mint azt említettem, a GABI jelenleg nem hozzáférhető, de érdemes a kutatócsoport honlapját figyelni későbbi fejleményekért: http://fonetikaitanszek.elte.hu/index.php/kutatas/gabi.

Irodalom

Bernstein Ratner, N., MacWhinney, B. 2016. Your laptop to the rescue: Using the Child Language Data Exchange System archive and CLAN utilities to improve child language sample analysis. *Seminars in Speech and Language* 37. 74–84. https://psyling.talkbank.org/years/2016/ssl/nan.pdf

Bernstein Ratner, N., MacWhinney, B. 2018. TalkBank resources for psycholinguistic analysis and clinical practice. In: Pareja-Lora, A., Blume, M., Lust, B. (szerk.). *Development of linguistic linked open data resources for collaborative data-intensive research in the language sciences*. Cambridge, MA: MIT Press. https://psyling.talkbank.org/years/2019/RatnerMacW.pdf

Bodor P., Barcza V. 2007. Acquisitioin of diminutives in Hungarian. In: *The acquisition of diminutives: A cross-linguistic perspective*. Amsterdam: Benjamins. 231–263.

Bóna J. 2017. GABI – Gyermeknyelvi beszédadatbázis a kutatásban. In: Bóna J. (szerk.). *Új utak a gyermeknyelvi kutatásokban*. Budapest: ELTE Eötvös Kiadó. 35–50.

Chang, F. 2017. *The LuCiD language researcher's toolkit* [Computer software]. http://gandalf.talkbank.org:8080/

MacWhinney, B. 1974. *How Hungarian children learn to speak*. Unpublished doctoral dissertation. Berkeley: University of California.

Mátyus K., Orosz Gy. 2014. MONYEK: Morfológiailag egyértelműsített óvodai nyelvi korpusz. *Beszédkutatás* 22. 237–245.

Orosz Gy., Mátyus K. 2014. An MLU estimation method for Hungarian transcripts. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. *Text, Speech, and Dialogue, of Lecture Notes in Computer Science*. Vol. 8655. 173–180.

Réger Z. 1986. The functions of imitation in child language. *Applied Psycholinguistics* 7/4. 323–352.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., Wintner, S. 2010. Morphosyntactic annotation of CHILDES transcripts. *Journal of Child Language 37*/3. 705–729. DOI: 10.1017/S0305000909990407. https://psyling.talkbank.org/years/2010/jcl-sagae.pdf

Theakston, A. L., Lieven, E. V. M., Pine, J. M., Rowland, C. F. 2001. The role of performance limitations in the acquisition of verb-argument structure: an alternative account. *Journal of Child Language* 28. 127–152.

Vakula T., Váradi V. 2017. Gyermeknyelvi hangfelvételek rögzítésének és lejegyzésének tapasztalatai. In: Bóna J. (szerk.). *Új utak a gyermeknyelvi kutatásokban*. Budapest: ELTE Eötvös Kiadó. 51–64.

A big data kihívás a bölcsészettudományokban: néhány digitális bölcsészeti kutatási eszköz bemutatása

Péter Róbert
egyetemi docens
Szegedi Tudományegyetem Bölcsészet- és Társadalomtudományi Kar, Angol-Amerikai Intézet, Angol Tanszék
robert.peter@ieas-szeged.hu

Absztrakt

Az előadás célja, hogy bemutasson néhány, digitális bölcsészeti kutatások során használt, jól ismert eszközt (pl. Google N-Gram Viewer, Bookworm, Voyant, Juxta Commons), valamint a Szegedi Tudományegyetemen az elmúlt évben kifejlesztett – és fejlesztés alatt lévő – két webszolgáltatást (TANIT és AVOBMAT). A TANIT (Text ANalysIs Tools) rendszer célja, hogy magyar nyelvű szövegek számítógépes nyelvészeti feldolgozásával dokumentumok összehasonlító elemzéséhez szükséges statisztikákat kigyűjtsön. Ez a webszolgáltatás létező nyelvtechnológiai elemzőlánc kimenetére épülő aggregált statisztikákat számít ki, témamodelleket épít, és ezeket olyan formátumban adja át a digitális bölcsész felhasználónak, aki utána programozói ismeretek nélkül is fel tudja ezt használni kutatásaiban. Az AVOBMAT (Analysis and Visualization of Bibliographic Metadata and Texts) segítségével nagy mennyiségű metaadatot és szöveget tudunk elemezni és vizualizálni. Az AVOBMAT-ba saját fájlokat és könyvtári repozitóriumokat tölthetünk fel, többek között Zoteróból exportált csv és rdf, valamint EP3 xml formátumokban. A feltöltés után tudjuk az adathalmazt szűkíteni fazettás, összetett és CCL kereséssel. A metaadatokat számtalan módon tudjuk interaktív módon vizualizálni, amelynek segítségével új, eleddig ismeretlen összefüggéseket és trendeket fedezhetünk fel digitális bölcsészeti elemzések során. Az AVOBMAT az egyszerű vizualizációk segítségével tudja például modellezni szerzők, kiadók, kulcsszavak kapcsolatát és időbeni eloszlását. A mesterséges intelligenciás módszereket és technológiákat használó összetett vizualizációs funkciók között szerepel a szerző–kiadó–(eladó) hálózatok interaktív vizualizációja, valamint női–férfi szerzők automatikus azonosítása és modellezése 55 nyelven. Az AVOBMAT működését a szegedi Acta repozitórium és brit sajtócikkek elemzésével fogom demonstrálni.

kulcsszavak: digitális bölcsészet, big data, TANIT, AVOBMAT