KorKorpusz: kézzel annotált, többrétegű pilotkorpusz építése

Vadász Noémi

Nyelvtudományi Intézet
Budapest, Benczúr utca 33.
e-mail: vadasz.noemi@nytud.hu

Kivonat A cikk egy többrétegű, kézzel annotált korpuszt ismertet, bemutatja annak elemzési rétegeit – különös tekintettel az anafora- és koreferenciaannotációra – és az építés fázisait, valamint felvillantja a felhasználási lehetőségeket. A korpusz szabadon elérhető és felhasználható, az építéshez használt eszközök és dokumentációik, valamint az annotálási útmutatók biztosításával pedig lehetőség nyílik annak további szövegekkel történő bővítésére.
Kulcsszavak: korpusz, annotálás, anafora, koreferencia

1. Háttér

A KorKorpusz tervezésekor a jelenleg létező legnagyobb magyar koreferenciakorpusz nyújtott inspirációt[1]. A SzegedKoref (Vincze és mtsai, 2015) a Szeged Korpusz (Csendes és mtsai, 2005) egy részét felhasználva készült, újsághíreket és iskolai fogalmazásokat láttak el koreferenciaannotációval. A legutóbbi publikáció alapján a SzegedKoref (Vincze és mtsai, 2018) 400 szöveget, 4 021 mondatot és 55 763 tokent tartalmaz. A szövegekben 2 456 anaforikus láncot[2] jelöltek meg.

Mi szükség van a SzegedKoref mellett még egy magyar koreferenciakorpuszra? Ez a kérdés több irányból is megközelíthető. A kézzel annotált, jó minőségű adat nagyon értékes erőforrás, és minél több van belőle, annál jobb. A cikkben ismertetett KorKorpusz összes elemzési rétege – a SzegedKorefhez hasonlóan – kézzel ellenőrzött minőségű, így nem csak az anafora- és koreferenciaannotáció hasznosítható belőle, hanem a többi nyelvi elemzés is. A két korpusz elemzési rétegei között azonban vannak különbségek, míg a SzegedKoref az MSD morfológiai kódkészlet[3] (Erjavec, 2004) egy feature-value formában megfogalmazott verzióját[4] használja morfológiai címkekészletként, addig a KorKorpusz morfológiai

[1] A magyar nyelvű koreferenciakorpuszok között meg kell említeni a Miháltz és mtsai (2007) és Miháltz (2012) tudásalapú koreferenciafeloldó rendszerének kiértékeléséhez használt korpuszokat. Ezek általános iskolai történelemkönyvből vett szövegekből állnak, amelyekben kézzel annotálták a különböző típusú anaforikus- és koreferenciakapcsolatokat. A korpuszokon egy annotátor dolgozott és a fent hivatkozott cikkek részletesen leírják az annotált típusokat, ám a korpuszok sajnos nem hozzáférhetőek.

[2] Az anaforikus láncok magukban foglalják a névmási anaforikus kapcsolatokat és a koreferenciaviszonyokat is.

[3] http://nl.ijs.si/ME/Vault/V3/msd/msd.pdf

[4] https://github.com/dlt-rilmta/panmorph/blob/master/panmorph_conll.pdf

rétege emMorph[5] (Novák és mtsai, 2017) és UD-kompatibilis[6] morfológiai címkéket tartalmaz. Egy másik különbség, hogy a SzegedKoref összetevős elemzést, míg a KorKorpusz dependenciaelemzést használ szintaktikai elemzésre. Végül a KorKorpusszal szemben a SzegedKoref nem tartalmaz tövesítést.

A fent említett különbségek ellenére elképzelhető a két korpusz együttes használata, így a SzegedKoref kb. 55 ezer tokenje kiegészülhet a KorKorpusz mindenkori tartalmával. Ehhez csupán az eltérő formátumú koreferenciaannotációt kell egységes formára hozni. Ugyanakkor fontos megemlíteni, hogy a KorKorpusz tervezésekor bizonyos elméleti kérdésekben másképp döntöttünk, mint a SzegedKoref-ben (például a KorKorpuszban az infinitívusz alanya is megjelenik zéró névmásként, beillesztettük a zéró létigéket és az elliptált igéket, jelöljük az általános alanyokat is, különválasztottuk az anaforikus kapcsolatokat a koreferenciaviszonyoktól stb).

A korpusz tervezésekor szem előtt tartottunk a könnyű elérhetőséget, használhatóságot és továbbfejleszthetőséget. A SzegedKoref engedélykérés után kutatási és oktatási célokra felhasználható, míg a KorKorpusz az összes dokumentáció és útmutató társaságában CC-BY-4.0 licensszel elérhető, így bárki továbbfejlesztheti és publikálhatja az eredményeit.

2. Anafora és koreferencia

Az információkinyerés és a kivonatolás területein számos olyan feladat van, amelyek megoldásához anafora- vagy koreferenciafeloldásra van szükség. Egy ilyen kapcsolatokat is tartalmazó korpusz hasznos erőforrás, akár tanítóanyagként, akár a kiértékelés során. Ám a szöveget átszövő kapcsolatok annotálása is nagyobb kihívást jelent.

A koreferencia és az anafora fogalma gyakran összemosódik, hiszen mindkét kapcsolattípus feloldása feltétele a szöveg pontos interpretációjának. Szem előtt kell tartani ugyanakkor a köztük lévő fontos különbségeket. Amint van Deemter és Kibble (1999) rávilágít, nem mindig világos, hogy az egyes korpuszok esetében pontosan mit értenek koreferencia alatt. Felhívja a figyelmet arra is, hogy míg a koreferencia szimmetrikus és tranzitív kapcsolat, addig az anafora nem, viszont az anafora kontextusfüggő.

A kétféle kapcsolat közötti különbség a KorKorpusz anafora- és koreferenciaanotálásánál is megmutatkozott, a részleteket lásd a 4.7. és a 4.8 fejezetekben.

3. A korpusz főbb adatai

A KorKorpusz az `e-magyar` újabb verziójának (Indig és mtsai, 2019) keretében használt formátumot[7] követi, amelyben a tokenek soronként szerepelnek és a

[5] `https://e-magyar.hu/en/textmodules/emmorph_codelist`
[6] `https://github.com/dlt-rilmta/panmorph/blob/master/panmorph_ud.pdf`
[7] `https://github.com/dlt-rilmta/xtsv`

mondatokat üres sor választja el egymástól. A különböző elemzési rétegek tabbal elválasztott oszlopokban kapnak helyet, az oszlopok sorrendje nem kötött, azt a fájl első sorában található fejléc határozza meg. A korpusz jelenleg 95 dokumentumot, 1 436 mondatot és 31 492 tokent tartalmaz, amelybe beleszámítanak az írásjelek és a zéró elemek is[8].

Jelenleg két forrásból gyűjtött szövegeket tartalmaz a korpusz, amelyeket az OPUS gyűjteményéből (Tiedemann, 2012) válogattuk. A szövegek egy részét a magyar Wikipédiáról gyűjtöttük, másrészt a GlobalVoices hírportál[9] magyar nyelvre lefordított hírei közül válogattunk. A KorKorpusz örökli ezeknek a forrásoknak a nyílt hozzáférhetőségét.

A korpusz az építéséhez készített eszközökkel és dokumentációikkal, valamint az annotálási útmutatókkal együtt az alábbi GitHub repozitóriumban érhető el: `https://github.com/vadno/korkor_pilot`.

4. A korpuszépítés lépései

A korpusz tervezésekor a munkát egy feldolgozó láncolatként képzeltük el. Célunk volt, hogy minél több lépést automatizáljunk és emberi munkát csak az eszközök kimenetének javításához használjunk.

Bizonyos elemzési lépésekhez az `e-magyar` második verzióját (Indig és mtsai, 2019) használtuk, amelyre ezentúl a cikkben munkanevén, `emstv`-ként hivatkozunk. Mivel az `emtsv` egy szövegfeldolgozó pipeline, ahol egy adott elemzési lépés kimenete a következő lépés bemenetét képezi, így nem lenne hatékony csupán a – jelen bemutatott korpusz szempontjából – legutolsó lépés után kézzel javítani a kimenetet, hiszen addigra a korábbi lépésekben keletkezett hibák hógolyóként még több hibát görgetnének maguk előtt. Így többlépéses kézi javítást alkalmaztunk, amely ugyan idő- és munkaigényes feladat, viszont könnyebben kontrollálható. Az `emtsv` kimenetét két körben volt szükséges ellenőrizni és javítani, ezután a saját eszközeink (zérónévmás-beszúró, anaforafeloldó) kimenetét is ellenőrizni kellett.

Az alábbi felsorolás tartalmazza az egyes elemzési és ellenőrzési lépéseket. Zárójelben a használt eszközök neve jelenik meg (a saját fejlesztésű eszközök **vastag betűvel** kiemelve).

1. szöveggyűjtés
2. elemzés (emtsv/emToken, emtsv/emMorph, emtsv/emTag)
3. formátumátalakítás (**saját szkript**)
4. kézi ellenőrzés (Google Spreadsheets)
5. formátumátalakítás (**saját szkript**)
6. elemzés (emtsv/emDep)
7. formátumátalakítás (emtsv/emCoNLL)
8. kézi ellenőrzés (WebAnno)

[8] Mivel az annotálási munka jelenleg is folyik, az aktuális méretet lásd a korpusz repozitóriumában.

[9] `https://hu.globalvoices.org`

9. zéró létigék és igei ellipszisek kézi beillesztése (szövegszerkesztő)
10. zéró névmások beillesztése (**saját szkript**)
11. automatikus névmási anaforafeloldás (**saját szkript**)
12. kézi ellenőrzés és koreferenciaannotálás (Google Spreadsheets)
13. formátumátalakítás (**saját szkript**)

A kézi ellenőrzést igénylő munkafázisok esetében az annotátorok munkaidő-nyilvántartást vezettek, ahol nem csak azt rögzítették, hogy mely fájlokkal végeztek, hanem azt is, hogy az adott fájl ellenőrzésekor milyen nehézségekbe ütköztek. Ezen kívül azt is követtük, hogy az egyes fájlok ellenőrzése – az egyes elemzési szinteken – mennyi időt vett igénybe, ezáltal a korpusz további bővítésének költségei is kalkulálhatóak. Az 1. táblázat azt mutatja, hogy átlagosan hány percet vett igénybe egy dokumentum ellenőrzése a különböző elemzési szinteken.

	perc/dokumentum
morfológiai egyértelműsítés ellenőrzése	0:24:13
függőségi elemzés ellenőrzése	0:29:23
anaforák beillesztésének ellenőrzése	0:34:22

1. táblázat. A kézi ellenőrzéshez szükséges idő a különböző elemzési lépések után.

Érdemes szem előtt tartani a tényt, hogy az első néhány fájl ellenőrzése mindig több időt vett igénybe. Az annotátorok minden felmerülő problémát, nehézséget jeleztek, így az annotálási útmutató is finomodott, egyre pontosabb és világosabb iránymutatást biztosított, így a munka is felgyorsult.

A folyamatok ellenőrzéshez egy összevető program is készült, az emDiff[10], amely lehetővé teszi az eltérő tokenizálású szövegek simítását[11] és az egyes oszlopok tartalmának összevetését. Ennek köszönhetően nem csak a több annotátor által annotált ugyanazon szövegek összevetésére alkalmas, hanem annotátorok közötti egyetértés számítására[12] is. Végül az annotátorok által ellenőrzött végleges verzió és az egyes elemzők által produkált kimenetek is összevethetőek, így ezeknek az elemzőknek a teljesítménye is kiértékelhető a program segítségével. A program az `emtsv` moduljaként is futtatható.

Az egyes lépések között a fájlok formátuma többször is változik, ahol a rákövetkező lépés bemeneti fájlformátuma eltér. A folyamat legutolsó lépése a fájlok átalakítása az `emtsv` által használt rugalmas formátumra.

A következőkben részletezzük a korpusz építésének egyes lépéseit.

[10] `https://github.com/vadno/emdiff`

[11] a Python `difflib` csomagjával (`https://docs.python.org/3/library/difflib.html`)

[12] az `nltk. metrics` csomaggal (`https://www.nltk.org/api/nltk.metrics.html`)

4.1. Szöveggyűjtés és előkészítés

A fent említett forrásokból több mondatot tartalmazó szövegeket gyűjtöttünk, hiszen az anafora- és koreferenciaviszonyok mondathatárokon is átívelnek. A szövegek hossza 5 és 27 mondat között, a mondatok hossza 3 és 71 token között van (az írásjeleket külön tokennek számolva).

Az összegyűjtött szövegeket `emtsv` elemzőeszköz számára megfelelően kellett előkészíteni. Bár a pilotkorpusz szövegei standard helyesírásúak voltak, szükségesnek tartottunk minden szöveget átnézni. A Wikipédia és a Global Voices szövegeiben is bőven találtunk nem standard helyesírású szöveget, ezeket a nyers szövegek átolvasása során kézzel javítottuk. A szövegeket egyszerű szövegfájlokban (`txt` kiterjesztéssel, UTF-8 karakterkódolással) tároltuk el, szövegenként külön fájlban.

4.2. emToken, emMorph, emTag

Az `emtsv` megfelelő moduljai (emToken (Mittelholcz, 2017), emMorph (Novák, 2014; Novák és mtsai, 2016; Novák, 2003) és emTag (Orosz és Novák, 2012, 2013)) segítségével történő elemzés kimenete egy négy oszlopot tartalmazó fájl, aminek a formátumát röviden már a 3. fejezet ismertette. Az oszlopok tartalma a következő: token, emMorph kimenet, egyértelműsített tő, egyértelműsített morfológiai címke. Az emMorph kimenet a szó összes lehetséges elemzését – különböző címkeformátumokban – és az azokhoz tartozó tövet tartalmazza. Az egyértelműsített morfológiai címke az emMorph morfológiai címkekészletét használja.

4.3. Kézi ellenőrzés

Az első kézi ellenőrzés a tokenizálás, a tövesítés és az egyértelműsített morfológiai címke ellenőrzését jelentette. A feladat elvégzéséhez a be- és kimeneti formátum rugalmassága, az ergonomikus és könnyen használható felület, könnyű elérés, a verziókövetés és a kollaboratív felület kritériumainak a Google Spreadsheets felelt meg.

A három nyelvész annotátor az előkészített (összegyűjtött, átnézett, `emtsv`-vel tokenizált, morfológiailag elemzett és egyértelműsített), valamint a Google Spreadsheets formátumára igazított szövegeket szerkesztette. A feltételes formázások célja, hogy vezessék az annotátor szemét a munka során, kiemeljék a potenciálisan javítandó elemeket, valamint visszacsatolást nyújtsanak a javításról.

Az annotátorok a táblázatban a tokenizálás és a tövesítés javítása mellett az egyértelműsített morfológiai címkét (tehát az emTag kimenetét) javították. Ehhez az emMorph által produkált összes lehetséges morfológiai elemzés rendelkezésre állt, amelyek közül ki kellett választani a helyes elemzést a tővel együtt. Ha a morfológiai elemzések közül egyik sem volt helyes, akkor kézzel is meg lehetett adni a megfelelő elemzést.

A tokenizálási hibák javítására kézzel beírható parancsokat határoztunk meg, amelyeket aztán az exportált `csv` feldolgozásakor automatikusan értelmez egy

szkript és azoknak megfelelően módosítja a tokenizálást (sort töröl vagy sort szúr be az annotátor által megadott tartalommal).

Az exportált `csv` feldolgozásakor a tokenizálási javításokra vonatkozó parancsok értelmezése mellett az `emtsv` formátumára történő visszaalakítás is megtörtént. A kézi javítás után a szöveg tehát pontosan ugyanúgy néz ki, mint javítás előtt, különbség a kijavított mezőkben van csupán.

A szövegek egy részét az összes annotátor ellenőrizte, így ezeken a szövegeken kiszámolható az annotátorok közötti egyetértés. A javítás során az annotátorok arra vonatkozóan nem kaptak útmutatót, hogy az emMorph címke által megjelenített derivációkat és a szóösszetételeket hogyan kezeljék, hiszen a függőségi elemző már egy csak az inflexiós jegyeket megjelenítő címkekészlet alapján dolgozik (a részleteket lásd a 4.4. alfejezetben). Hogy az ebből fakadó különbségeket ne vegyük figyelembe az annotátorok közötti egyetértés számolásakor, már az emMorph-ról erre az inflexiós jegyeket tartalmazó készletre konvertált címkéket használtuk. A 4 315 tokennyi, mindhárom annotátor által ellenőrzött szövegre kapott eredmény Krippendorff alfában (Artstein és Poesio, 2008) kifejezve: 0,976.

Az annotációs útmutató és az ahhoz tartozó kiegészítő dokumentum, a feltételes formázásokat tartalmazó Google táblázat, valamint a táblázat formátumára alakító szkript elérhető a korpusz repozitóriumában.

4.4. emDep

A címkék kézi javítása után a szövegek szintaktikai elemzését az `emtsv` függőségi elemző modulja végez el. A függőségi elemzés előtt az emMorph címkét át kell alakítani a függőségi elemző modul számára emészthető UD-kompatibilis morfológiai címkére, amely megegyezik a Szeged Dependency Treebank (Vincze és mtsai, 2010) és az `emtsv` függőségi elemzője[13] (Zsibrita és mtsai, 2013), az emDep címkekészletével[14]. A továbbiakban erre a címkekészletre UD-ként hivatkozunk[15]. Ezt a konverziót is az `emtsv` egy modulja[16] végzi el. Az UD-re való konvertáláskor a derivációra vonatkozó egyes információk elvesznek. Azzal, hogy az emMorph címkék lettek kézzel javítva, és nem a már UD-re konvertált címke, többletmunkát végeztünk. Ezzel a többletmunkával azonban elértük, hogy a korpusz ezen rétege, a morfológiai egyértelműsítés is kézzel ellenőrzött minőségű.

A következő lépésben a javított és az `emtsv` formátumának megfelelően visszaalakított szövegeket az `emtsv` függőségi elemzőjével elemeztük.

4.5. Kézi ellenőrzés és zéró létigék

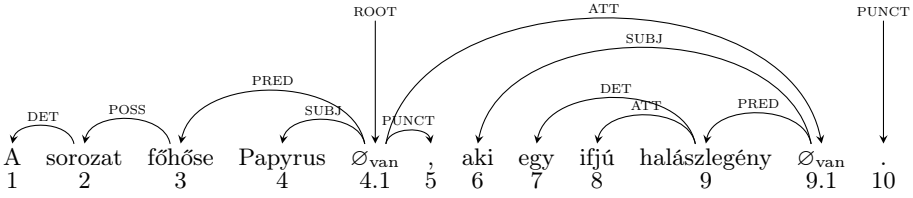
A javításhoz a WebAnno[17] általános célú, webalapú eszközt (Eckart de Castilho és mtsai, 2016) használtuk, hiszen a legtöbb fontos szempontnak megfelelt.

[13] `http://e-magyar.hu/hu/textmodules/emdep`
[14] A címkekészleteket Vadász és Simon (2019) részletesen ismerteti.
[15] `https://github.com/dlt-rilmta/panmorph/blob/master/panmorph_ud.pdf`
[16] `https://github.com/vadno/emmorph2ud`
[17] `https://webanno.github.io/webanno`

Drag-and-drop módszerrel használható, az elemzés különböző fázisaiban lévő dokumentumok is feltölthetőek, nem csak annotálásra, hanem javításra is használható. Olyan kiegészítő funkciókkal is bír, mint a több annotátor által kezelt dokumentumok összevetése, a munka egyszerű nyomonkövetése és az annotátorok közötti egyetértés különböző mérőszámok alapján történő automatikus kiszámolása. Az eszköz rugalmasnak mondható, hiszen saját elemzési rétegeket is megfogalmazhatunk. A WebAnno egy szerveren fut, az annotátorok pedig a megszokott böngészőjükön keresztül használhatják. A függőségi elemzés után az `emtsv` moduljaként működő konverterrel[18] alakítottuk át a kimenetet a WebAnno számára emészthető CoNLL-U formátumra. Az ellenőrzést három nyelvész annotátor végezte.

Használat közben azonban mégis felmerültek problémák ezzel az eszközzel kapcsolatban. Bár a tokenizálási hibák javítására már korábban volt lehetőség, mégis előfordult, hogy a függőségi elemzés kimenetének javításakor találkoztunk ilyen hibákkal. Sajnos a WebAnno felületén token törlésére vagy beszúrására nincs mód, így ezt a problémát egy utófeldolgozó lépésben kellett kezelni.

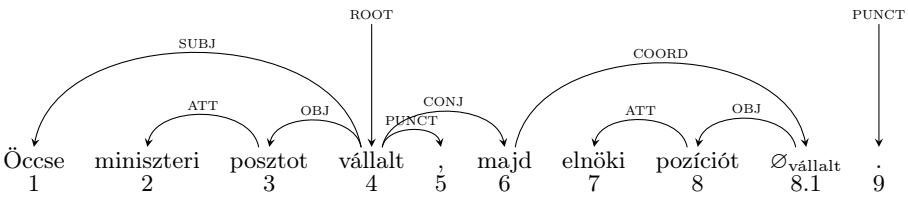
Az esetleges tokenizálási hibák javításával egyidőben a zéró létigék és az elliptált igék beszúrása is megtörtént. A zéró létigék beszúrását kézzel végeztük el azokban a mondatokban, amelyekben nem volt finit ige. A zéró létigék új tokenként kerülnek a fájlba arra a helyre, ahol múlt időben testes létigeként jelennének meg, saját kombinált indexet kapnak, ami a zéró létigét megelőző elem ID-jéből képződik. Az 1. példában egy olyan mondat látható, ahol a függőségi fába két zéró létigét is be kellett illeszteni.



1. ábra: Az összetett mondat fölérendelt tagmondatának zéró létigéje alá van rendelve az alárendelt mellékmondat zéró létigéje. A második sorban a kiosztott indexek láthatók a zéró létigék kombinált indexével együtt.

Az igei ellipsziseket is jelöltük a korpuszban, hiszen gyakran találkoztunk olyan tagmondatokkal, amelyekben az elliptált ige hiánya miatt nem lehetett megfelelő anyacsomóponthoz kötni az egyes bővítményeket. A zéró létigékhez hasonlóan kézzel illesztettük a mondatfába az elliptált finit igéket. Az elliptált ige a zéró létigéhez hasonló, kombinált indexet kapott. A 2. példában egy olyan mondat látható, ahol a függőségi fába egy elliptált igét kellett beilleszteni.

[18] `https://github.com/vadno/emconll`



2. ábra: A mellérendelés első mondatában szereplő ige a második mondatban testetlenül van jelen. Ezért egy zéró alakot illesztettünk be a függőségi fába, így a második mondatban szereplő vonzat már kapcsolódni tud a saját testetlen igéjéhez.

A korpusz 463 beillesztett zéró létigét és 25 beillesztett elliptált igét tartalmaz.

4.6. A zérónévmások beillesztése

A zérónévmásokat egy saját szkript, az emZero[19] illeszti be, amelynek bemenete a tokenizált és (javított) tövesítéssel, morfológiai egyértelműsítéssel és függőségi elemzéssel ellátott szöveg. Egyszerű szabályok mentén végzi az elemek beillesztését és a szabályok alkalmazása során különböző elemzési rétegek tartalmára támaszkodik (tő, morfológiai címke, függőségi elemzés).

A program a következő helyekre illeszt be zérónévmást:

- finit ige alanyának, ha annak nem volt testes alanya
- határozott ragozású finit ige tárgyának, ha annak nem volt testes tárgya
- birtok birtokosának, ha annak nem volt testes birtokosa
- ragozott és ragozatlan infinitívusz alanyának

A zérónévmások beillesztése után a mondatfában plusz ágak jelennek meg. A zéró elemek is saját ID-t kapnak, a tsv-be pedig az alany az ige után, a tárgy az ige (és a zéró alany) után, a birtokos pedig a birtok után kerül és egy kombinált ID-t kap, ami az őt megelőző elem ID-jéből és a zéró elem szintaktikai szerepének rövidítéséből (SUBJ, OBJ, POSS) áll. A zéró elemek szófaja névmás (PRON), a morfológiai jegyeik között pedig az ige vagy a birtok alapján kiszámolható szám és személy jegyek jelenhetnek meg.

A program az `emtsv` moduljaként is futtatható.

4.7. Névmási anaforák beillesztése

A következő lépésben a névmási anaforikus kapcsolatokat is egy szabályalapú szkript szúrja be. A program megkeresi a névmásokat, majd a mondatban szereplő többi szó szófaji, morfológiai és szintaktikai információira támaszkodva egyszerű szabályok alapján dönt.

[19] `https://github.com/vadno/emzero`

A szkript jelenleg csak a személyes névmások előzményét keresi meg, a többi típust kézzel kell beilleszteni. A személyes névmások előzménykeresésének egyszerű algoritmusát Pléh és Radics (1976) alapján dolgoztuk ki. Az algoritmus az alany antecedensének keresésekor például az alábbihoz hasonló szabályok alapján dönt:

1. ha az ige alanya zéró névmás és az ige ragozása az előző mondat igéjének ragozásához képest nem változott, akkor az alany antecedense az előző mondat igéjének alanya
2. ha az ige alanya mutatónévmás, akkor annak antecedense az előző mondat nem alanyi argumentuma

4.8. Kézi ellenőrzés és koreferenciaannotálás

Az automatikusan beillesztett zéró névmások és névmási anaforák ellenőrzését, valamint a koreferenciaannotálást négy nyelvész annotátor végezte.

Számos annotációs eszköz található, amelyek segítségével lehet anafora- és koreferenciaéleket annotálni a szövegekben (pl. WebAnno, brat[20] (Stenetorp és mtsai, 2012), TrEd[21] stb.). Vannak olyan eszközök is, amelyek az annotáció javítására használhatók és például CoNLL-U formátumban képesek feldolgozni az adatot. Legjobb tudomásunk szerint olyan eszköz azonban nem áll rendelkezésre, amely minden fontos kritériumunknak megfelelt volna, és emellett a zéró elemek kezelésére is alkalmas lenne (mert például a CoNLL-U formátum a hivatalos formátumleírás[22] alapján nem teszi lehetővé, hogy zéró elemek szerepeljenek a függőségi fában).

Az automatikusan beillesztett zérónévmások és anaforikus kapcsolatok ellenőrzését, valamint a koreferenciakapcsolatok beillesztését így ismét feltételes formázásokkal ellátott Google Spreadsheets táblázatokban végeztük el. Az anaforikus- és koreferenciakapcsolatokat két oszlopban kellett jelölniük az annotátoroknak, egyikben annak az elemnek az ID számát kellett megadni, amellyel a visszautaló elem kapcsolatban áll, a másikban pedig a kapcsolat típusát. A korpuszban az alábbi anaforikus kapcsolattípusokat jelöltük (zárójelben a korpuszban szereplő jelölésükkel):

- személyes (**prs**)
- mutató (**dem**)
- kölcsönös (**recip**)
- visszaható (**refl**)
- vonatkozó (**rel**)
- birtokos (**poss**)

Az automatikus névmási anaforákat beillesztő program a személyes névmások előzményén kívül nem ad számot a többi névmásról és azokról a kapcsolatokról,

[20] `http://brat.nlplab.org`
[21] `http://ufal.mff.cuni.cz/tred`
[22] `https://universaldependencies.org/format.html`

amelyekben ezek szerepelnek. Ilyen az általános alany szerepben álló zéró névmás, amelynek referenciája nehezen megragadható (**vastag betűvel** kiemelve az általános referenciájú alannyal rendelkező igéket (1. példák).

(1) a. ... a Kínai Kommunista Párt egyik volt vezetője, akit hazaárulás miatt **elítéltek** ...

b. 1883-ban **említették** először az orthodox hitközségnek adományozott területként ...

Hasonlóak azok esetek, amikor a szöveg írója megszólítja az olvasót (2. példa, ahol **vastag betűvel** kiemeltük azokat az igéket, amelyeknek alanya az író vagy az olvasó). Ez a típus nem gyakori a hírszövegekben vagy a Wikipédia-szövegekben, ugyanakkor a korpusz más műfajú szövegekkel (pl. szépirodalom, személyes szövegek) történő bővítésénél gyakrabban előfordulhat.

(2) A születésnapi ajándékoknak is nagyon **örülünk**, ha **szeretnéd** támogatni a munkánkat, **küldj nekünk** adományt, vagy **vegyél** egyet az NSA-s karácsonyi **üdvözlőlapjaink** közül, amelyet a Creative Time-nál dolgozó **barátaink** terveztek.

Új címkéket vezettünk be ezekre az esetre: az **arb** az általános alany, az **addr** a címzett, a **speak** a beszélő/író referenciáját jelöli. Azzal, hogy bevezetjük a szöveg szereplői közé a beszélőt és a címzettet, jelölni tudjuk, ha a szövegben szereplő névmások ezen szereplők valamelyikével koreferens antecedensre utalnak vissza.

Az egyes koreferencakorpuszokban, így a SzegedKoref annotációjában is különböző típusú koreferenciakapcsolatokat (pl. ismétlés, variáció, szinonima, hipernima, hiponima és holonima stb.) jelölnek. Az annotálás tervezése során és a szövegeket megfigyelve azonban számos nehézségbe ütköztünk ezekkel a típusokkal kapcsolatban. A korpusz annotációja így csupán kétféle koreferenciatípust különböztet meg.

A **coref** címkével jelölt koreferenciatípus magában foglalja az összes olyan koreferenciakapcsolatot, amely két azonos referenciájú elemet köt össze, így például az ismétlést, a szinonimát, hiper- és hiponimát. A **holo** címkével jelölt kapcsolattípus pedig azt jelenti, hogy a két szó referenciája között rész-egész viszony áll fenn, pontosabban a második szó referenciája része az első szóénak.

Míg a koreferenciakapcsolatok előzménye (amellyel közös referenciájuk van vagy referenciájuk között rész-egész viszony áll fenn) mindig testes szó, addig a (testes vagy zéró) névmások előzménye (antecedense) lehet tartalmas szó, illetve testes vagy testetlen névmás. Ennek megfelelően az anaforikus és koreferenciakapcsolatok nem folyamatos láncot képeznek a szövegen át, hanem elágazásokat, kitérőket is tartalmaznak.

A 2. táblázat összefoglalja, hogy a korpusz összesen hány visszautalást tartalmaz az egyes kapcsolattípusokból. Mindemellett az ellenőrzés végén a korpusz 2 346 zéró alanyt, 260 zéró tárgyat és 914 zéró birtokost tartalmaz.

kapcsolat	előfordulás
prs	1 497
dem	147
recip	11
refl	18
rel	447
poss	0
arb	316
speak	5
addr	1
coref	1 582
holo	202

2. táblázat. Az anaforikus és koreferenciakapcsolatok előfordulása a KorKorpuszban.

4.9. Nehézségek

A koreferencia annotálásakor számos nehézséggel találtuk szembe magunkat, amelyek kezelésére a szakirodalom sem tudott megnyugtató választ kínálni. A 3. példa azt a problémát illusztrálja, amikor a referens állapota megváltozik (itt: meghal). Vajon a holttest koreferens az emberrel?

(3) Három hónap telt el az **újságíró házaspár**, Sagar Sarwar és felesége, Meherun Runi meggyilkolása óta. A **holttesteket** már exhumálták is, hogy megismételjék a boncolást.

A 4. példa azt a nehézséget szemlélteti, amikor egy szó előzménye, amellyel koreferens, több tagból áll.

(4) **Papyrus** bátor és megmenti **Thèti-Chèri-t**. **A két egymásra lelt barát** küldetést kap az istenektől, hogy védelmezzék meg a fáraót.

A két egymásra lelt barát egyszerre koreferens *Papyrussal* és *Thèti-Chèrivel*, sőt, csak az egyikükhöz kötni feltétlenül hibás volna. Ugyanakkor a jelenlegi annotációs séma alapján csak egyetlen előzményhez lehet kötni. Az se sokat javítana a helyzeten, ha mellérendelés állna fenn *Papyrus* és *Thèti-Chèri* között. Ugyan ebben az esetben már ábrázolható lenne a koreferenciakapcsolat *a két egymásra lelt barát* és a mellérendelés feje között, viszont a feloldása többértelmű lenne, hiszen nem lehetne eldönteni, hogy a teljes mellérendelő szerkezetre, vagy csak a fejében lévő elemre történt-e a visszautalás.

Az ezekhez hasonló problémás esetek kezelésére külön irányelveket kell kidolgozni, ám az ezekkel kapcsolatos döntések még előttünk állnak.

5. A pilotkorpusz hasznosíthatósága

A KorKorpusz már a jelenlegi pilot fázisban is több célra hasznosítható. A kézzel annotált, jó minőségű adat értékét nem lehet eléggé hangsúlyozni, legyen szó bármely elemzési feladatról. A KorKorpusz kézzel javított elemzési rétegei lehetővé teszik, hogy megvizsgáljuk, hogy a korpuszépítéshez használt egyes eszközök milyen minőségű elemzést biztosítottak, így nemcsak a KorKorpusz építéséhez újonnan készült előelemző eszközök, hanem az `emtsv`-ben használt modulok teljesítménye is kimérhető.

Az emTag (Orosz és Novák, 2012, 2013) teljesítményét a 4. fejezetben ismertetett emDiff segítségével vizsgáltuk meg azon a 122 fájlon, amelyekben az egyértelműsítés eredményét kézzel javítottuk. Összevetettük az egyértelműsített tő és az egyértelműsített morfológiai címke mezők tartalmát. Az eredményeket a 3. táblázat tartalmazza.

	pontosság
tő	98,15%
morfológiai címke	95,40%

3. táblázat. Az emTag teljesítménye pontosságban (*accuracy*) kifejezve.

Az eredmény alapján elmondható, hogy kevés esetben kellett kézzel kivajítani a címkéket. Az egyes hibatípusok csoportosítása a későbbiekben segítséget nyújthat az emTag kimenetének automatikus javításában is.

6. További tervek

A KorKorpusz további fejlesztésének két iránya van: egyrészt a rendelkezésre álló eszközök és dokumentációk segítségével a korpusz további szövegekkel való kibővítésével, másrészt az egyes munkafolyamatok még könnyebbé és ergonomikusabbá tételével. Távlati tervek között szerepel az elkészített eszközök további javítása, valamint a koreferenciakapcsolatok automatikus beillesztésének kidolgozása.

Köszönetnyilvánítás

Hálámat fejezem ki az annotátoraimnak, Bencze Norbertnek, Bognár Ivettnek, Fegyó Kingának és Fodor Grétának, akik a monoton feladatok elvégzése mellett friss ötleteikkel folyamatosan inspiráltak.

Hivatkozások

Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Computational Linguistics 34(4), 555–596 (2008)

Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S.M., Hartmann, S., Gurevych, I., Frank, A., Biemann, C.: A web-based tool for the integrated annotation of semantic and syntactic structures. In: Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH). pp. 76–84. The COLING 2016 Organizing Committee, Osaka, Japan (Dec 2016), `https://www.aclweb.org/anthology/W16-4011`

Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference, TSD 2005. pp. 123–131. Springer, Karlovy Vary, Czech Republic (2005)

van Deemter, K., Kibble, R.: What is coreference, and what should coreference annotation be? In: Coreference and Its Applications. pp. 90–96 (1999)

Erjavec, T.: MULTEXT-East Morphosyntactic Specifications. Version 3.0 (May 2004), http://nl.ijs.si/ME/Vault/V3/msd/html/

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: emtsv – egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019)

Miháltz, M.: Tudásalapú koreferencia- és birtokviszony-feloldás magyar szövegekben. Általános Nyelvészeti Tanulmányok 24, 151–166 (2012)

Miháltz, M., Naszódi, M., Vajda, P., Varasdi, K.: NP-koreferenciák feloldása magyar szövegekben a magyar wordnet ontológia segítségével. In: Tanács, A., Csendes, D. (szerk.) V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016). pp. 138–146. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged (2007)

Mittelholcz, I.: `emToken`: Unicode-képes tokenizáló magyar nyelvre. In: Vincze, V. (szerk.) XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017). pp. 70–78. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2017)

Novák, A.: Milyen a jó Humor? In: I. Magyar Számítógépes Nyelvészeti Konferencia. pp. 138–144. SZTE, Szeged (2003)

Novák, A.: A new form of Humor – Mapping constraint-based computational morphologies to a finite-state representation. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (may 2014)

Novák, A., Rebrus, P., Ludányi, Zs.: Az `emMorph` morfológiai elemző annotációs formalizmusa. In: XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017). p. (jelen kötetben). Szeged (2017)

Novák, A., Siklósi, B., Oravecz, Cs.: A new integrated open-source morphological analyzer for Hungarian. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk,

J., Piperidis, S. (szerk.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)

Orosz, Gy., Novák, A.: PurePos 2.0 – an open source morphological disambiguator. In: Sharp, B., Zock, M. (szerk.) Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science. pp. 53–63. Wroclaw (2012)

Orosz, Gy., Novák, A.: PurePos 2.0: a hybrid tool for morphological disambiguation. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013. pp. 539–545. INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria (Sep 2013), https://www.aclweb.org/anthology/R13-1071

Pléh, Cs., Radics, K.: „Hiányos mondat", pronominalizáció és a szöveg. Általános Nyelvészeti Tanulmányok 11(1), 261–277 (1976)

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: Brat: A web-based tool for nlp-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 102–107. EACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)

Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (szerk.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (2012)

Vadász, N., Simon, E.: Konverterek magyar morfológiai címkekészletek között. In: Berend, G., Gosztolya, G., Vincze, V. (szerk.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 99–112. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019)

Vincze, V., Hegedűs, K., Farkas, R.: SzegedKoref: kézzel annotált magyar nyelvű koreferenciakorpusz. In: Tanács, A., Varga, V., Vincze, V. (szerk.) XI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 312–322. SZTE TTIK Informatikai Tanszékcsoport (2015)

Vincze, V., Hegedűs, K., Sliz-Nagy, A., Farkas, R.: SzegedKoref: A Hungarian coreference corpus. In: Proceedings of the 11th Language Resources and Evaluation Conference. European Language Resource Association, Miyazaki, Japan (2018)

Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., Csirik, J.: Hungarian dependency treebank. In: Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (szerk.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (2010)

Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: Proceedings of RANLP. pp. 763–771 (2013)