

**Számítógépes anafora- és koreferenciafeloldás
magyar nyelvű szövegeken**

Vadász Noémi

témavezető: Prószéky Gábor

Pázmány Péter Katolikus Egyetem

Bölcsészettudományi Kar

Nyelvtudományi Doktori Iskola

2024

Köszönetnyilvánítás

Köszönöm a témavezetőmnek, Gábornak, hogy bízott bennem és lehetőséget adott, hogy számítógépes nyelvesszé váljak. Köszönöm a bírálóimnak, Gyurinak és Tibornak a kritikákat, amik segítettek a dolgozat végső változatának elkészítésében. A dolgozatban több olyan eredményt is ismertetek, amely közös munkával jött létre, ezért külön szeretném megköszönni társszerzőimnek, Eszternek, Noéminek és Bencének, hogy együtt dolgozhattunk.

Köszönettel tartozom azoknak pécsi és pázmányos tanárainknak, akik elindítottak az utamon és azoknak a kollégáimnak, akik segítettek, támogattak, tanítottak, jó ötleteket adtak, fontos kérdéseket tettek fel, bíztak bennem, és időről időre helyretették az önbizalmamat. Nagyon hálás vagyok a Nyelvtechnológiai kutatócsoportnak, akik 2018-ban befogadtak és az első megbeszélésünkön elkészült az `osztalybemutatkozo_tanacsok.txt` fájl. Az itt kapott ötletek mind megvalósultak, eredményük olvasható a dolgozatban. Hálával tartozom Tamásnak, hogy lehetővé tette, hogy építsem a korpuszomat. Köszönöm Marcinak, hogy először megemlítette nekem a Winograd sémákat, Ivánnak, hogy sugározta a higgadtságot, Bálintnak, hogy mindig a legérdekesebb kérdéseket tette fel, Áginak, hogy taposta előttem az utat, Balázsnak, hogy segített megvalósítani az elképzeléseimet, Bencének, hogy a legpozitívabban állt hozzám, Noéminek, hogy minden közös munkánk lelkesítő volt, Kingának, hogy támogatott, amikor elment a kedvem, Lacinak és Verának, hogy visszahozták a kedvemet. Nem tudom eléggé megköszönni Eszternek a szakmai útmutatást, a mentorálást és a barátságot.

A legnagyobb hálával Andrisnak és Palinak tartozom.

Tartalomjegyzék

1. Bevezetés	1
1.1. Motiváció	1
1.2. A dolgozat felépítése és a tézisek	2
1.3. Fogalmak	7
I. Erőforrások	12
2. Erőforrások anafora- és koreferenciafeloldáshoz	13
2.1. Pro-drop nyelvek koreferenciakorpuszai	16
2.2. Magyar koreferenciakorpuszok	17
3. Egy magyar koreferenciakorpusz: KorKor	18
3.1. Magyar előzmény: a SzegedKoref	19
3.2. A korpusz főbb adatai	21
3.3. A korpuszépítés folyamata	21
3.3.1. Az <code>xt sv</code> fájlok összevetése: <code>emDiff</code>	23
3.3.2. Szöveggyűjtés	26
3.3.3. Morfológiai elemzés és egyértelműsítés az <code>emt sv</code> elemzővel	26
3.3.4. A morfológiai címkék kézi javítása	26
3.3.5. Az egyértelműsítő kiértékelése	29
3.3.6. Függőségi elemzés az <code>emt sv</code> elemzővel	30
3.3.7. A függőségi elemzés kézi javítása	31
3.3.8. A függőségi elemző kiértékelése	33
3.3.9. A zéró létigék beillesztése	35
3.3.10. A zérónévmás-beszűrő	36
3.3.11. A névmási anaforikus kapcsolatok beillesztése	37
3.3.12. Kézi ellenőrzés és koreferenciaannotálás	37
3.3.13. Formátumok	40
3.4. Nehézségek	42

Tartalomjegyzék

3.5.	Konverterek magyar morfológiai címkekészletek között	44
3.5.1.	Magyar morfológiai címkekészletek	45
3.5.2.	A konverterek	47
3.5.3.	Kiértékelés	50
3.5.4.	Eredmények és diszkusszió	51
3.6.	A KorKor a CorefUD gyűjteményben	54
3.7.	Összefoglalás	56
4.	NYTK-NerKor	58
4.1.	Előzmények	59
4.2.	Műfajok	60
4.3.	Források	60
4.4.	Az annotáció formátuma	62
4.5.	A morfológiai annotálás első üteme	62
4.5.1.	A szövegek kiválasztása	63
4.5.2.	Előfeldolgozás	63
4.5.3.	Az annotációs folyamat és felület	64
4.5.4.	Annotátorok közötti egyetértés	64
4.6.	A morfológiai elemzés második üteme	65
4.7.	Összefoglalás	68
5.	Adatbázisok a névmási anaforafeloldáshoz	70
5.1.	A Winograd-sémák	71
5.1.1.	A Winograd-séma kihívás mint az új Turing-teszt	71
5.1.2.	Benchmark adatbázisok Winograd-sémákkal	73
5.2.	A Winograd-sémák fordításai	74
5.2.1.	Francia	74
5.2.2.	Portugál	75
5.2.3.	Kínai	76
5.3.	További adatbázisok	76
5.3.1.	Definite Pronoun Resolution Dataset	76
5.3.2.	Winogrande	77
5.3.3.	Wino-X	78
5.3.4.	XWINO	79
5.3.5.	Pronoun Disambiguation Problems (PDP)	79
5.4.	A magyar Winograd-sémák és hasonló adatbázisok	80
5.4.1.	HuWS: magyar Winograd-sémák	81

Tartalomjegyzék

5.4.2.	HuWNLI: A magyar WNLI adatbázis	82
5.4.3.	PWS: Parallel Winograd-sémák	83
5.4.4.	HAPP: Hungarian Ambiguous Pronoun Problems	84
5.5.	Összefoglalás	85
II.	Eszközök	86
6.	Anafora- és koreferenciafeloldó eszközök	87
6.1.	Módszerek az anafora- és a koreferenciafeloldásban	88
6.2.	Magyar anafora- és koreferenciafeloldók	89
7.	Zérónévmás-beszúrás két módszerrel	91
7.1.	Zérónévmások és testetlen alanyok	91
7.2.	Szabályalapú zérónévmás-beszúrás: emZero	95
7.2.1.	Kiértékelés	96
7.3.	Zérónévmás-beszúrás nyelvmodell finomhangolásával	97
7.3.1.	A finomhangolás	98
7.3.2.	Kiértékelés	99
7.4.	Diszkusszió	100
8.	Szabályalapú anaforafeloldás magyar szövegeken	101
8.1.	Szabályalapú anaforafeloldó: DepEll	101
8.1.1.	Az ANAGRAMMA működési alapelvei	101
8.1.2.	Névszói frázisok az ANAGRAMMA elemzőben	102
8.1.3.	A Pléh-Radics algoritmus	106
8.1.4.	Koreferenciafeloldás az ANAGRAMMA elemzőben	108
8.1.5.	Az alany keresése	108
8.1.6.	A tárgy keresése	111
8.1.7.	A többi vonzat keresése	112
8.1.8.	Többértelmű anafora	113
8.1.9.	Diszkusszió	113
8.2.	A KorKor építésekor használt szabályalapú anaforafeloldó	114
8.2.1.	Az anaforafeloldó működése	114
8.2.2.	Kiértékelés	115
8.2.3.	Diszkusszió	118
9.	Koreferenciafeloldás BERT-tel	120
9.1.	A felhasznált korpuszok	121

Tartalomjegyzék

9.2. Szófajalapú előszűrés	123
9.3. Finomhangolás	124
9.4. Vizualizáció	125
9.5. Klaszterezés	127
9.6. Kiértékelés	127
9.6.1. A szófajalapú előszűrés kiértékelése	127
9.6.2. A koreferenciafeloldás kiértékelése	128
9.6.3. Diskusszió	129
10. Anaforafeloldás a ChatGPT-vel	131
10.1. A ChatGPT és a Winograd-sémák	132
10.2. A kísérlet (első kör)	132
10.3. A válaszok kategorizálása	134
10.4. A válaszok lehetséges okai	135
10.5. Eredmények és diskusszió	141
10.6. A következetesség vizsgálata (második kör)	142
10.7. Diskusszió	144
11. Összefoglaló	145
12. Summary	148
13. Felhasznált irodalom	151

1. fejezet

Bevezetés

Egyenes sorban álltak a repülőgépek. Előttük a pilóták, feszes vigyázzban.

Orrukon vidáman ragyogott az acél propeller.

Lőrincze Lajos: Nyelvőrségen, XIV. Téves kapcsolás. 1968

1.1. Motiváció

Elméleti nyelvészeti háttérű számítógépes nyelvészként olyan témát kerestem a doktori kutatásaimhoz, ami egy érdekes nyelvi jelenséget foglal magába és emellett nyelvtechnológiai szempontból is kihívást jelent. A számítógépes anafora- és koreferenciafeloldás éppen ilyen, hiszen emellett, hogy elméleti nyelvészeti szempontból is kutatott és izgalmas terület, számítógépes szemszögből is sokféleképpen megközelíthető.

Az anafora- és koreferenciakapcsolatok feloldása a magas szintű nyelvfeldolgozási feladatok közé tartozik, ebből pedig két dolog következik. Egyrészt továbbra is sok a tennivaló a témában, így az anafora és a koreferencia jelensége aktívan foglalkoztatja a nyelvtechnológus közösséget. A Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC) 2023-ban a hatodik konferenciáját¹ rendezte meg a témában, ez az esemény az egyik legnagyobb fóruma a koreferencia és anafora számítógépes kezelésével kapcsolatos nemzetközi eredményeinek. A CRAC teret ad olyan éppen kibontakozó projekteknek is, amelyek nemzetközi standardok létrehozását célozzák,

¹<https://sites.google.com/view/crac2023/>

1.2. A dolgozat felépítése és a tézisek

mint például a különböző nyelvű, koreferenciaannotációkat tartalmazó korpuszok összegyűjtésével és harmonizálásával foglalkozó CorefUD², valamint a Universal Dependencies³ mintájára létrehozott Universal Anaphora⁴ kezdeményezés, amely nyelvfüggetlen anafora- és koreferenciaannotációs séma elkészítésén dolgozik.

A fenti állításból következik másrészt az is, hogy ezeknek a kapcsolatoknak a helyes feloldása mutatója lehet annak, hogy egy gépi modell mennyire teljesít jól a nyelvértés területén. A szövegeket behálózó kapcsolatok feloldásához nem elegendő pusztán lexikális vagy grammatikai információkra támaszkodni, a kontextus értéke, pragmatikai ismeretek, világismeret és következtetési képességek is szükségesek hozzá. A legújabb nagy neurális nyelvmodellekkel kapcsolatban az az elvárásunk, hogy a nyelvet ezekkel a magas szintű aspektusaival együtt dolgozzák fel, éppen ezért ezen nyelvmodellek teljesítményét mérő tesztek (az ún. *benchmark* adatbázisok, pl. (Wang et al., 2018)) általában tartalmaznak az anaforák helyes feloldását mérő feladatokat is.

Dolgozatomban két szemszögből vizsgálom az anafora és a koreferencia jelenségeit: az erőforrások és az eszközök szemszögéből. Az erőforrások alapanyagként vagy üzemanyagként szolgálnak bizonyos nyelvtechnológiai feladatok megoldására készített eszközök számára, így a doktori munka keretében készített eszközök megkívánták az erőforrások elkészítését is.

A fejezetekben ismertetett megoldások között szerepelnek szabályalapú, a nyelvi jelenségek grammatikai viselkedésén alapuló megoldások, de kitérek az anafora neurális nyelvmodellek értékelésében betöltött szerepére is. Ennek a kettősségnek az oka az, hogy a sikeres anaforafeloldás kulcsa egyrészt a nyelvi jelenség mögött húzódó grammatikai szabályszerűségek sikeres feldolgozása, másrészt pedig a nyelv megértéséhez szükséges világismeret, következtetési képesség és közös tudás.

1.2. A dolgozat felépítése és a tézisek

A dolgozat első felében olyan erőforrásokat ismertetek, amelyeket hozzáadott emberi munkával készítettem. A kézi annotációval készülő erőforrásokkal kapcsolatban elsősorban az foglalkoztat, hogy hogyan lehet a legjobbat kihozni egy gondosan elkészített alapanyagból legyen szó kisebb, specifikusabb korpuszok elkészítéséről különböző kutatási kérdések megválaszolására, vagy tanító- és tesztanyag előállításáról hagyományos vagy neurális gépi tanulási megoldások számára. Az erő-

²<https://ufal.mff.cuni.cz/corefud>

³<https://universaldependencies.org/>

⁴<https://universalanaphora.github.io/UniversalAnaphora/>

1.2. A dolgozat felépítése és a tézisek

források ismertetésekor kitérek az erőforrás megtervezésével, a munkafolyamat felállításával, vagy az annotátorok munkájának hatékonyabbá tételével és a kézi annotáció minőségének kiértékelésével kapcsolatos kérdésekre. A dolgozatban lehetőségem van több olyan erőforrást is bemutatni, amelyek az anafora és a koreferencia jelenségével kapcsolatosak, de kitérek egy további, a magyar nyelvtechnológia számára hasznos korpuszra is, amelynek építésekor a morfológiai annotáció elkészítéséért voltam felelős.

Az általam készített erőforrások között elsőként egy kézzel annotált korpuszt ismertetek. A KorKor korpusz a hagyományos nyelvi elemzési rétegek (egyértelműsített morfológiai elemzés, függőségi mondatelemzés) mellett anafora- és koreferenciaannotációt tartalmaz, minden elemzési rétege kézzel ellenőrzött minőségű. A korpusz kétféle formátumban is elérhető, az erőforrás szabadon felhasználható. A reprodukálhatóság jegyében az erőforrás mellett az építés teljes munkafolyamatát, a felhasznált szkripteket és az annotálási útmutatókat is elérhetővé tettem, ami más korpuszok építéséhez is segítséget nyújthat. A KorKor korpusz alkalmasnak bizonyult arra, hogy bekerüljön a CorefUD gyűjteménybe (Novák et al., 2022) is, ami standardizált formátumba rendezi a különböző nyelvekre készült koreferenciakorpuszokat is. A fejezethez az alábbi tézisek tartoznak:

1. tézis: Többretegű, kézzel ellenőrzött minőségű korpuszt építettem, amelynek fő annotációs rétege az anafora- és koreferenciaannotáció. A munka eredményeként létrejött korpusz, a KorKor mellett nemcsak a munkafolyamat pontos dokumentációját és az annotálási útmutatókat publikáltam, hanem elérhetővé tettem a korpuszépítéshez használt eszközöket is, amelyeket szintén magam fejlesztettem. A dokumentációk és eszközök lehetővé teszik a korpusz bővítését, vagy alapját képezhetik más típusú annotációval ellátott korpuszok elkészítésének is. A KorKor – bár eltérő annotációs sémákat és címkekészleteket használ –, összevontan is használható előzményével, a SzegedKoref korpuszsal (Vincze et al., 2015). A dolgozat fejezeteiben több példát is láthatunk a két korpusz összevont alkalmazására. A tézist alátámasztó publikáció: (Vadász, 2020; Vadász, 2022).

2. tézis: Társszerzővel feltérképeztem és egységes dokumentációval láttam el a legfontosabb magyar morfológiai címkekészleteket, valamint konvertereket készítettem bizonyos morfológiai címkekészletek között. A kutatást és a dokumentációkat társszerzőmmel együtt végeztem, a konvertereket magam implementáltam. A konverterek közül az emmorph2ud és az annak alapján készült emmorph2ud2 konverterek az `emt sv` (Indig et al., 2019a; Simon et al., 2020) keretrendszerében is használhatók. Az emmorph2ud konverter használata megkerülhetetlen az egyértelműsítő és a függőségi elemző moduljai között, hiszen ez a konverter állítja elő az `emTag` kimenetét az `emDep` bemenetének megfelelő címkeformátumra. A tézist alátámasztó publikáció: (Vadász és Simon, 2019).

1.2. A dolgozat felépítése és a tézisek

3. tézis: A KorKor korpuszt alkalmassá tettem, hogy beilleszkedjen a CorefUD gyűjteménybe, amelynek célja, hogy összegyűjtse és harmonizálja a különböző nyelvekre készült koreferenciakorpuszokat mind az annotációs sémát, mind pedig a formátumot tekintve. A korpusz így nagyobb láthatóságot kapott, az egységes annotációs sémának és formátumnak hála pedig más nyelvek koreferenciakorpuszaival is összevethető lett. A CorefUD projekt publikációja (Nedoluzhko et al., 2022) támasztja alá ezt a tézist.

A KorKor projekt eredményeit a társszerzővel készített NYTK-NerKor korpusz építéskor is hasznítani tudtuk: egyrészt alapanyagként használtuk fel a KorKor anyagát, másrészt a KorKor építési folyamatát, az építéskor készült eszközöket és útmutatókat is fel tudtuk használni. Az NYTK-NerKor korpusz rövid ismertetése mellett részletesebben kitérek az egyértelműsített morfológiai elemzés annotációs rétegének elkészítésére és a munkafolyamat során szerzett tapasztalatokra. A fejezet az alábbi tézishez kapcsolódik:

4. tézis: Társszerzővel együtt elkészítettük az NYTK-NerKor korpuszt, ami az egymillió tokenes méretével jelenleg a legnagyobb magyar névelemannotált korpusz. A korpusz elemzési rétegei gold standard minőségűek. A korpusz építése során társszerzőm volt felelős a névelemannotáció elkészítéséért, én pedig az egyértelműsített morfológiai elemzési réteg megtervezéséért és elkészítéséért. A morfológiai elemzési réteg elkészítéséhez a KorKor korpusz építéskor kidolgozott munkafolyamat releváns szakaszára és az ahhoz kapcsolódó eszközökre támaszkodtam, valamint a KorKor teljes anyagát beépítettem az NYTK-NerKor korpuszba. A tézist alátámasztó publikáció: (Simon és Vadász, 2021).

Egy másik erőforrás, amit a doktori disszertáció ismertet, egy társszerzővel készített sémagyűjtemény-készlet. Elkészült a Winograd-sémák (Levesque et al., 2012) magyar fordítása, valamint egy párhuzamos gyűjtemény, ami a Winograd-sémák elérhető fordításait tartalmazza hét nyelven. Emellett elkészült a Definite Pronoun Resolution Dataset (Rahman és Ng, 2012) 1 882 mondatának magyar fordítása is. A munka nem pusztán fordítási feladat volt, minden egyes séma esetében figyelni kellett arra, hogy az eredményül kapott magyar megfelelőben megőrződjön a Winograd-sémákra jellemző szerkezeti többértelműség. A fejezet az alábbi tézishez kapcsolódik:

5. tézis: Társszerzővel együtt elkészítettem a Winograd-sémák magyar fordításait és néhány további hasonló erőforrást. A sémagyűjtemények hiánypótlók, hiszen eddig nem létezett hasonló magyar erőforrás. A többértelmű névmási anafora feloldásának feladata jó indikátora lehet annak, hogy egy nyelvmodell mennyire képes megérteni a nyelvet, hiszen a feloldáshoz a grammatikai szerkezetek felismerésén túl szükség van lexikális ismeretekre, világismeretre és következtetési képességre is, éppen ezért volt szükség ezeknek az erőforrásoknak az elkészítésére, amelyek alkalmasak a

1.2. A dolgozat felépítése és a tézisek

nagy neurális nyelvmodellek kiértékelésére. A tézist alátámasztó publikáció: (Vadász és Ligeti-Nagy, 2022).

A dolgozat második felében olyan eszközöket mutatok be, amelyek az anafora- vagy a koreferenciafeloldás feladatát, vagy a hozzájuk köthető alfeladatok egyikét, a zérónévmások beillesztését oldják meg. Az általam készített, a zérónévmások beillesztését végző program egyszerű szabályokon alapul és működésekor az elemzett szöveg többi elemzési rétegére támaszkodik, tehát morfológiai és szintaktikai információkra. A zérónévmások beillesztésére egy másik módszerrel, neurális nyelvmodell, a huBERT (Nemeskey, 2021) finomhangolásával is kísérletet tettem. A fejezet az alábbi tézist fejti ki:

6. tézis: Zérónévmás-beszűrőt készítettem magyar nyelvre szabályalapú és neurális módszerekkel. A magyarra eddig nem létezett olyan eszköz, amely a zérónévmások beillesztését végezte volna el. A szabályalapú zérónévmás-beszűrőt a KorKor építéséhez előannotáció elkészítésére fejlesztettem ki, majd az elkészült korpuszt tanító- és kiértékelőadatként egy neurális megoldás elkészítésére használtam fel. A szabályalapú zérónévmás-beszűrőt emZero néven az `emt.sv` keretrendszerében is használható. A tézist alátámasztó publikáció: (Simon et al., 2020).

Egy másik, a dolgozatban bemutatott szabályalapú algoritmus a névmási anafora antecedenskeresését végzi. Szintén morfológiai és szintaktikai információkra támaszkodik, a mögöttes szabályrendszer alapja pedig az ún. Pléh-Radics algoritmus (Pléh és Radics, 1976). Az algoritmus eredetileg az AnaGamma (Prószéky és Indig, 2015) elemzőrendszer keretei között készült, később ezt az algoritmust fejlesztettem tovább, és az ez alapján készült szkriptet használtam a KorKor korpusz építése során az előannotáció elkészítésére. A fejezethez tartozó tézis:

7. tézis: A Pléh-Radics algoritmus alapján elkészítettem egy anaforafeloldó algoritmust, amely az AnaGamma elemzőrendszer működési alapelveinek is megfelel, majd az algoritmust implementáltam is. Az AnaGamma kereteibe illeszthető algoritmus a humán szövegfeldolgozás főbb jellemzőit veszi figyelembe. Az algoritmus alapján készített szabályalapú program bizonyos névmások antecedenstét azonosítja az előzményben. A program a KorKor korpusz releváns annotációs rétegének előannotálására készült, hogy meggyorsítsa és megkönnyítse a humán annotátorok munkáját. Az anaforafeloldó elkészítését az indokolta, hogy ugyan több magyar megoldás is fellelhető volt a szakirodalomban, sajnos egyik sem volt hozzáférhető. A tézist alátámasztó publikációk: (Vadász, 2017, 2020; Vadász, 2022).

A koreferenciafeloldás feladatára egy neurális megoldást is ismertetek, amelyet társszerzőmmel a huBERT finomhangolásával készítettünk. A finomhangoláshoz a két elérhető magyar nyelvű korefe-

1.2. A dolgozat felépítése és a tézisek

renciakorpuszt, a SzegedKoref korpuszt (Vincze et al., 2018) és a disszertációban bemutatott KorKor korpuszt egységesítve használtuk. A fejezet az alábbi tézishez tartozik:

8. tézis: Társ szerzővel együtt neurális anaforafeloldót fejlesztettünk. Az anaforafeloldó elkészítéséhez rendelkezésünkre állt megfelelő mennyiségű és minőségű tanítóanyag (a KorKor és a SzegedKoref korpusz), valamint finomhangolható mély neurális nyelvmodell (huBERT). Ennek ellenére mégsem létezett korábban neurális koreferenciafeloldó magyar nyelvre. A megoldásunk ezt a hiányt pótolja. A tézist alátámasztó publikáció: (Vadász és Nyéki, 2023).

Végül ismertetek egy kísérletet, amiben azt vizsgáltam, hogy mennyire alkalmas a ChatGPT a magyar nyelvű szövegek esetében az anaforafeloldás feladatára. A kísérletben a fent említett, a doktori kutatás keretében elkészített sémafordításokat használtam. A fejezethez tartozó tézis:

9. tézis: Kísérleteket végeztem azzal kapcsolatban, hogy a ChatGPT mennyire teljesít jól a többértelmű anaforák előzményének megtalálásában. A többértelmű anaforák sikeres feloldása megmutathatja, hogy a nyelvmodell milyen nyelvértési képességekkel rendelkezik. A feladatot a Turing-teszt alternatívájaként is szokták emlegetni, hiszen a sikeres anaforafeloldáshoz világismeretre és következtetési képességre is szükség van. A kísérleteimben a ChatGPT válaszadási stratégiáit is vizsgáltam amellett, hogy mennyire képes sikeresen feloldani a többértelmű névmási anaforákat. A fentiek mellett arra is kerestem a választ, hogy milyen következetesen kapunk helyes választ a feltett kérdéseinkre. A tézist alátámasztó publikáció: (Vadász, 2023).

A doktori munka keretében készült erőforrások és eszközök mind a magyar nyelvre készültek. Ennek egyik oka, hogy a magyar nyelvtechnológiában ugyan többen foglalkoznak már ezekkel a jelenségekkel, azért még akadt megoldásra váró feladat. A másik oka pedig az, hogy a magyar mint pro-drop nyelv kifejezetten izgalmas terep a névmási anaforák vizsgálatára.

Céлом, hogy az általam készített erőforrások és eszközök valóban használhatók legyenek mások számára, ezért külön figyelmet fordítottam az alapos dokumentációra és a hozzáférhetőségre. Szintén a használhatóságot segítő törekedtem arra, hogy az általam készített erőforrások elterjedt szten-derdekhez igazodjanak, az eszközeim pedig szintén bevett környezetekben és keretrendszerekben használhatók legyenek.

1.3. Fogalmak

1.3. Fogalmak

A dolgozatot a saját hozzájárulásom bemutatása előtt a számítógépes anafora- és a koreferenciafeloldás fontos fogalmainak tisztázásával kell kezdenem, emellett összegyűjtöttem néhány, a dolgozat megértéséhez szükséges számítógépes nyelvészeti fogalmat is. Fontos kiemelni, hogy a fogalmak meghatározása eltérhet a nyelvi jelenségek elméleti nyelvészeti meghatározásaitól. A számítógépes nyelvészetben belül az anafora- és koreferenciafeloldás információkinyerési feladatok, tehát a feloldó rendszereknek általában az a céljuk, hogy a szövegekben megjelöljék azokat az elemeket, amelyek között anaforikus- vagy koreferenciakapcsolat áll fenn. A feloldók sok esetben nem célozzák ezeknek a kapcsolatoknak az osztályozását, sem az anaforikus- és a koreferenciakapcsolatok szétválasztását, a korpuszok pedig gyakran szintén egy kalap alá veszik őket. A szakirodalomban gyakran előfordul, hogy a koreferencia alá tartozó fogalomként kezelik az anaforikus kapcsolatokat is.

Lássuk először az anaforikus- és a koreferenciakapcsolatok rövid definícióját. [Pléh \(1998\)](#) az 1.1. táblázatban látható visszaulástípusokat tartja számon.

zéró anafora	<i>A gyerek megölelte a tanárt. Leült és elmosolyodott.</i>
névmási anafora	<i>A gyerek megpuszilta a nénit. Ő meg elmosolyodott.</i>
ismétlés	<i>A gyerek meglátta a kutyát. A kutya ugatni kezdett.</i>
alá-, fölérendelés	<i>Laci vett egy pulit. A kutya sokat ugat.</i>
szinonímia	<i>Feri meglátta a rendőrt. A zsaru hirtelen megfordult.</i>
epitheton	<i>Ede elhagyta Marit. A disznó még el sem köszönt.</i>
valószínű rész	<i>A rendőr belépett a szobába. Az ablak nyitva volt.</i>
szükségszerű rész	<i>Tegnap lerobbant a kocsim. A fém romlott el.</i>
esetkeret	<i>A rabló lelőtte a pénztárost. A pisztoly a földön hevert.</i>

1.1. táblázat. Visszaulástípusok és egy-egy példa [Pléh \(1998\)](#) alapján. Anaforikus- vagy koreferenciakapcsolat a *dőlttel* szedett elemek között áll fenn.

A **koreferenciakapcsolat** esetén két nyelvi elem azonos referenciával bír, míg az **anaforikus kapcsolat** esetén egy saját referenciával nem bíró elem (pl. egy névmás) utal vissza egy szövegbeli előzményre. Az 1.1. táblázatban sorolt visszaulástípusok között az első két esetben anaforikus kapcsolat áll fenn a visszaulató elem (**anafora**) és előzménye (**antecedens**) között, a többi esetében a két elem között koreferenciakapcsolat van. Az antecedens lehet referáló vagy nem referáló elem is, tehát például egy másik névmás.

1.3. Fogalmak

van Deemter és Kibble (2000) így foglalja össze az anafora és a koreferencia közötti különbséget: míg a koreferencia egy ekvivalencia-reláció, addig az anaforikus kapcsolatok irreflexívek, nem szimmetrikusak és nem tranzitívak. Emellett az anafora interpretálása kontextusfüggő, a koreferenciáé pedig nem. Az anaforikus és a koreferenciális kapcsolatok egybeeshetnek, de egyik sem a másiknak alkategóriája.

Az anafora- és koreferenciafeloldás nyelvtechnológiai szempontból összetett feladat, több alfeladat tartozik hozzá, az alfeladatokat pedig különbözőképpen határozzák meg az egyes rendszerek attól függően, hogy milyen megközelítést alkalmaznak. A koreferenciafeloldást gyakran klaszterezési feladatként határozzák meg, ahol a szövegbeli elemeket kell aszerint osztályozni, hogy mi a valóságbeli referenciájuk. Ez alapján egy klaszterbe kerülnek azok az elemek, amelyeknek azonos a referenciája. Ez a megközelítés nem tesz különbséget az anafora és a koreferencia nyelvi jelenségei között, nem kategorizálja a különböző típusú relációkat az elemek között, ugyanakkor lehetővé teszi az egyelemű klaszterek felállítását is. Ez utóbbi azt jelenti, hogy minden olyan elem is kaphat klasztercímkét, ami a szövegben aktuálisan semelyik másik elemmel nem áll anaforikus- vagy koreferenciakapcsolatban. Ezt nevezhetjük **entitásalapú** (*entity-based*) megközelítésnek.

Egy másik lehetséges megoldás, ha nem az elemek referenciái alapján állítunk fel az osztályokat, hanem a visszautaló vagy a referáló elemek szövegbeli előzményeit keressük meg. Ekkor a visszautaló (pl. névmás) vagy referáló (pl. főnév) elemnél megjelöljük a lineáris sorrendben megelőző elemet, amire visszautal vagy amivel koreferens. Ennél a megközelítésnél csak azok az elemek kerülnek megjelölésre, amelyeknek van előzményük vagy velük koreferens másik elem a szövegben. Ezt **linkalapú** megközelítésnek (*link-based*) nevezhetjük. A linkalapú megközelítés esetében a szövegben szereplő kapcsolatok (**koreferencia**)**láncokat** alkotnak. Az entitásalapú megközelítés szemszögéből nézve az egy láncban szereplő elemek közös klaszterbe tartoznak, ezért a két megközelítés valamelyest átjárható. A linkalapú megközelítés azonban nem ad számot azokról az elemekről, amelyek az entitásalapú megközelítésben egyelemű klaszterek tagjai lennének.

Ha a vizsgált nyelvi jelenségek tulajdonságait és működését tekintjük, akkor az entitásalapú megközelítés inkább a koreferenciakapcsolatok ábrázolására alkalmas, hiszen a klaszterek jól kifejezik a koreferenciakapcsolatra jellemző szimmetrikus, tranzitív és reflexív tulajdonságokat. A linkalapú megközelítés ezzel szemben jól ábrázolja az anaforikus kapcsolatokra jellemző aszimmetrikus viszonyt a visszautaló elem és az antecedens között.

További különbség lehet az egyes annotációs sémák és a feloldók között például az is, hogy a kapcsolatokban résztvevő elemeket hogyan jelölik: a teljes összetevőt vagy csupán a frázis fejét jelölik-e meg. Az előbbi megoldás a szerkezeti többértelműségek szempontjából szerencsésebb, mert példá-

1.3. Fogalmak

ul egy mellérendelő szerkezet esetében kiderül, hogy a teljes mellérendelő szerkezet vesz részt egy koreferenciakapcsolatban, vagy csak a mellérendelés utolsó tagja.

A számítógépes anaforafeloldás esetében kevésbé jellemző, hogy kezelik a kataforákat – ahol a visszautaló elem megelőzi azt az elemet, amelyre utal (tehát a posztcendensét). Ugyanakkor fontos megjegyezni, hogy információkinyerési szempontból ezek is lényeges kapcsolatok a szöveget behálózó kapcsolatrendszerekben.

Néhány fontos további fogalomra nem találtam megfelelőt a magyar szakirodalomban, ezért egy rövid definíció megadása mellett megpróbálom elnevezni őket magyarul. A korpuszépítés, azon belül a korpuszannotáció során használt fogalom a **jelölt** (*markable*), ami azt az egységet jelenti, ami részt vehet koreferenciakapcsolatban. Ez az egység lehet egy token vagy egy tokenszekvencia is, függően a korpusz annotációs sémájától. Ennek a fogalomnak a párja a koreferenciafeloldás területén az **utalás** (*mention*), ami a szövegben potenciálisan koreferenciakapcsolatban részt vevő elemre használatos fogalom. A koreferenciafeloldás során az utalások kiválogatása (*mention detection*) külön alfeladatként is meghatározható.

A dolgozat megértéséhez kapaszkodót nyújthat néhány számítógépes nyelvészeti fogalom meghatározása is. A doktori munka egy része korpuszok építéséről szól, ezért a korpuszokhoz kötődő fogalmakkal kezdem. Gyakori, hogy ezeket a fogalmakat a magyar szakirodalom és a magyar számítógépes nyelvészet közössége is angolul használja, ezért több fogalom mellett feltüntettem az angol megfelelőt is. A **korpusz** olyan elektronikus formátumban tárolt, előre meghatározott alapelvek mentén válogatott szövegek gyűjteménye, amely szövegeket valamiféle nyelvi **annotációval** láttak el. Az annotációt gépi eszközökkel vagy humán **annotátorok** segítségével készítik el, és az annotációhoz **annotációs alapelvek** és **címkekészlet** (*tagset*) is tartoznak. Az alapelvek és a címkekészlet ismerete szükséges az annotált korpusz gépi és kézi feldolgozásához is.

A **gold standard** korpusz annotációját humán annotátorok készítik el. Az annotátorok az előre elkészített **annotációs útmutató** (*guideline*) alapján dolgoznak, hogy a kézzel előállított elemzés megfeleljen az annotációs sémának. Ideális esetben a korpusz szövegeit legalább két annotátor annotálja, a két annotátor által produkált annotációt közös döntések vagy egy felülbíró annotátor döntései alapján egységesíteni kell. Ennek megfelelően a kézzel annotált korpuszok előállítása igen erőforrásigényes feladat. Az annotációs útmutató teljességének, az annotátorok felkészültségének és az egyéb annotációs körülmények alkalmasságának – így összességében a kézzel annotált korpusz minőségének – a becslésére szolgál az **annotátorok közötti egyetértés** (*inter-annotator agreement*) mérése. Az annotátorok közötti egyetértést a korpusz építése során többször érdemes megmérni, alacsony értékek esetén pedig módosítani, bővíteni az útmutatót, javítani az annotációs körülményeket. A kézzel

1.3. Fogalmak

annotált korpuszokkal együtt közzétett annotátorok közötti egyetértési eredmények a korpusz felhasználói számára is informatívak, hiszen az annotáció megbízhatóságát mutatják.

A korpuszok erőforrásként szolgálnak a nyelvtechnológia számára. Ezeket az erőforrásokat egyrészt **tanítóanyagként** (*training data*) lehet használni felügyelt gépi tanulási módszerekkel készülő modellek építéséhez, vagy neurális modellek **finomhangolására**. Mindkét módszer lényege az, hogy a modell az annotált adatban összefüggéseket találjon az annotáció és a szöveg bizonyos jellemzői között, hogy ezen összefüggések birtokában képes legyen annotációval ellátni új, még nem látott adatot is. A korpuszok másik felhasználási módja, hogy **tesztanyagként** (*test data*) az ugyanolyan annotációt kibocsátó eszközök minőségét lehet rajtuk mérni, ezt **kiértékelésnek** (*evaluation*) nevezik. Ekkor a gépi annotációt vetik össze a kézi annotációval és azt vizsgálják, hogy milyen mértékben hasonlít rá. Különböző mérőszámokkal lehet megmutatni a hasonlóság mértékét. Az érthetőség kedvéért egyszerű címkézési feladatokon értelmezve a **pontosság** (*precision*) azt mutatja meg, hogy a gépi modell milyen mértékben volt képes a megfelelő címkét párosítani egy elemhez a címkével ellátott elemek közül, míg a **fedés** (*recall*) azt mutatja meg, hogy a modell mennyire volt képes megtalálni egy adott kategóriába tartozó összes elemet. A két mérőszám átlaga, az **F-mérték** akkor lesz magas, ha a modell mindkét szemszögből jól teljesít, tehát helyesen válogatja ki a címkézendő elemeket és képes őket a megfelelő címkékkal ellátni.

Gyakori, hogy a korpuszokat úgy teszik közzé, hogy a fenti két feladatnak megfelelő halmazokra, alkörpuszokra bontják őket, így azoknak az eszközöknek az eredményei, amelyeket az adott korpuszon tanítottak és értékelték ki, összevethetők lesznek. A korpusz legnagyobb része (általában legalább a 80%-a) szokott **tanítóanyagként** (*training set*) szolgálni, a maradék mennyiség **tesztanyagként**. A tesztanyag egy részét gyakran elkülönítik az eszköz fejlesztése közbeni kiértékelésekhez, ezt **fejlesztő adatnak** (*development set*) nevezzük, a végső kiértékeléshez félretett halmazt pedig kiértékelő halmaznak (*test set*).

Végül ismertetek néhány fájlformátumot, amelyek többször felmerülnek a dolgozatban, elsősorban az erőforrásokról szóló részben. Az egyik fontos fájlformátum az **xtsv**⁵, amely az *emtsv* (Indig et al., 2019b) keretében használt formátum. Az *xtsv* egy fejléces *tsv* fájl, amelyben a tokenek soronként szerepelnek és a mondatokat üres sor választja el egymástól. Az egyes annotációk a tokenek mellett megjelenő oszlopokban kapnak helyet, az annotációs rétegek helyét a fejléc határozza meg. Az *xtsv* formátumú adat bevezethető az *emtsv* eszközláncba, ugyanakkor más, nemzetközileg elterjedt formátumokra is konvertálható, például a Universal Dependencies (UD)⁶ projektből kiinduló

⁵<https://github.com/dlt-rilmta/xtsv>

⁶<https://universaldependencies.org/>

1.3. Fogalmak

nemzetközileg elterjedt sztenderd fájlformátumra, a **CoNLL-U** formátumra az `emtsv` keretrendszeren belül működő konverter segítségével. A CoNLL-U formátum az `xtsv` formátumhoz hasonlóan szintén soronként egy tokent tartalmaz, a mondatokat szintén üres sor választja el egymástól, valamint ez is egy oszlopos fájlformátum. Az annotációkat tartalmazó oszlopok sorrendje azonban rögzített, a lehetséges annotációk a UD útmutatóit és címkekészleteit szolgálják⁷, így ez a formátum elsősorban a UD gyűjteményében megjelenő korpuszok formátuma. A **CoNLL-U Plus** formátum a CoNLL-U kiterjesztése, annál rugalmasabb fájlformátum⁸. A szerkezete megegyezik az elődjével, ugyanakkor az eredeti CoNLL-U formátum oszlopainak bármely részhalmazát tartalmazhatja, valamint bármennyi további projektspecifikus annotációt. Az elemzési rétegek sorrendjét a fájl első sorában megjelenő komment tartalmazza.

⁷<https://universaldependencies.org/format.html>

⁸<https://universaldependencies.org/ext-format.html>

I. rész

Erőforrások

2. fejezet

Erőforrások anafora- és koreferenciafeloldáshoz

Ebben a fejezetben ismertetek néhány fontosabb erőforrást, amelyet az anafora- vagy a koreferenciafeloldás feladatához készítettek. A legismertebb angol vagy más nyelvű korpuszok ismertetése után bemutatom az eddig készült magyar erőforrásokat. Kimerítő felsorolásra és minden részletre kiterjedő leírásra nincs mód, azonban kiválasztottam néhány korpuszt, amelyek méretükkel vagy valamely egyéb tulajdonságukkal kitűnnek a többi korpusz közül. A 2.1. táblázat foglalja össze az itt sorra vett korpuszokat és néhány fontosabb tulajdonságukat.

A TüBa-D/Z treebank (Hinrichs et al., 2004) egy német nyelvű, 3 816 újsághírt, 104 787 mondatot és 1 959 474 tokent, valamint több elemzési réteget tartalmazó treebank. A korpusz kézzel annotált elemzési rétegekből (inflexiós morfológia, tövesítés, összetevős mondatelemzés, grammatikai funkciók, komplex névelemek, anafora- és koreferenciaviszonyok és GermaNet (Hamp és Feldweg, 1997) jelentésannotációk) és automatikusan generáltakból (függőségi mondatelemzés és főnévi csoportok) áll. A korpusz sajtószövegekből (Die Tageszeitung) áll. A TüBa-D/Z anafora- és koreferenciaannotációja alapján a jelöltek a definit főnévi csoportok és a különböző névmásfajták. A jelöltek között a következő lehetséges kapcsolattípusok állhatnak fent: koreferencia, anafora, katafora, kötés, rész-egész, példány és expletív. A koreferenciaannotáció útmutatója a honlapról letölthető, maga a korpusz egy

	méret	nyelv	műfaj
Tüba-D/Z	1 959 474 token	német	sajtó
WikiCoref	59 652 token	angol	Wikipedia
GAP	8 908 szöveg	angol	Wikipedia
PCC	540 000 token	lengyel	vegyes
ParCorFull	160 000 token	angol-német párhuzamos	TED előadások, hírek
OntoNotes 5.0	2 600 000 token	angol, arab, kínai	vegyes
NAIST Text corpus	38 384 mondat	japán	NA
AnCora-CO	808 718 token	spanyol, katalán	hírek
ParCorfFull 2.0	269 568 token	angol, német, portugál, francia	TED előadások, hírek
ZAC	35 000 token	brazíliai portugál	vegyes
SzegedKoref	55 763 token	magyar	hírek, fogalmazások

2.1. táblázat. A fejezetben ismertetett korpuszok és néhány fontos tulajdonságuk.

licenc elfogadása után tudományos célokra felhasználható. Azért emeltem ki ezt a korpuszt, mert példásan dokumentált, karbantartott honlapja¹ van és az annotációs útmutató² is elérhető.

A WikiCoref (Ghaddar és Langlais, 2016) az angol nyelvű Wikipedia szócikkeit tartalmazza. A szócikkek struktúrája, az interwiki linkek mind hozzájárulnak ahhoz, hogy a belőlük épülő koreferenciakorpusz hasznosítható legyen egy koreferenciafeloldó eszköz számára. A kézi annotálást automatikus előfeldolgozó lépések előzték meg, így az annotátoroknak az automatikus koreferenciafeloldás eredményét kellett kézzel kijavítaniuk. Az annotációs útmutató az OntoNotes korpusz koreferenciaannotációjának sémáján (Pradhan et al., 2007b) alapul, némi módosítással. A korpusz teljes egészében elérhető, letölthető, kiválóan dokumentált. 30 dokumentumot, 2 229 mondatot és 59 652 tokent tartalmaz.

A GAP (Balanced Corpus of Gendered Ambiguous Pronouns) korpusz (Webster et al., 2018) angol Wikipédia szövegeket tartalmaz, amelyekben 8 908 kétértelmű névmás–név párost annotáltak kézzel. A korpusz célja, hogy egyenlő arányban tartalmazzon névmásokat a két nyelvtani nemből, hogy olyan alkalmazásokat lehessen készíteni a felhasználásával, amelyen nem elfogultak az egyik nyelvtani nem irányába sem.

¹<https://uni-tuebingen.de/en/faculties/faculty-of-humanities/departments/modern-languages/departments-of-linguistics/chairs/general-and-computational-linguistics/ressources/corpora/tueba-dz/>

²<http://www.sfs.uni-tuebingen.de/resources/tuebadz-coreference-manual-2007.pdf>

Amint [Hendrickx és Hoste \(2009\)](#) rávilágítottak, a sajtószövegeken tanított koreferenciafeloldók teljesítménye drasztikusan csökken, ha más műfajú szövegeken, például szerkesztetlen kommentszövegeket alkalmaznak őket. Ezért nagyon értékesek azok a koreferenciaannotációval ellátott korpuszok, amelyek többféle doménből és műfajból tartalmaznak szövegeket. Ilyen többműfajú korpusz lengyel nyelvre a PCC (Polish Coreference Corpus) ([Ogrodniczuk et al., 2016](#)), amely a maga kb. 540 000 tokenes méretével a legnagyobb koreferenciakorpuszok között van. Két alkorpusza van, a 1 773 dokumentumból álló, átlagosan 250–350 szegmentumot tartalmazó „rövid szövegek” nagyobb szövegek kiragadott részletei (pl. napi újságok, magazinok, szépirodalom, használati utasítások, tudományos szövegek, parlamenti leiratok, Wikipedia, blogbejegyzések, kommentek, levelek, hirdetések, beszélt szövegek leiratai a médiából, párbeszédleiratok stb.). A „hosszú szövegek” különböző témájú cikkek (pl. jogi és gazdasági szöveg, sport, kultúra, tudományos szöveg) a Rzeczpospolita online újságból összegyűjtött korpuszból (Korpusz Rzeczpospolitej ([Weiss, 2002](#))). A korpusz szabadon elérhető, letölthető, alaposan dokumentált³. Rendelkezésre áll egy online felület is, ahol a brat-formátumú⁴ korpuszt lehet böngészni⁵.

A ParCorFull párhuzamos korpusz ([Lapshinova-Koltunski et al., 2022](#)) a ParCor 1.0 ([Guillou et al., 2014](#)) szövegeit felhasználva készült, angol-német párhuzamos korpusz, amelyben teljes koreferenciaalánccok vannak annotálva. Két műfajból tartalmaz szövegeket: TED előadások előre megírt szövegeit és EU Bookshop publikációkat. Az annotációt kézzel készítették, jelölték a névmások típusát és azt, hogy maga a névmás antecedens-e. A ParCorFull automatikus előannotálással készült, az annotálási útmutatóban hívják fel a figyelmet a rendszeres hibákra, amelyeket az annotátorok kézzel javítottak. Az útmutató a MUC (Message Understanding Conference) koreferenciafeladatának az útmutatója alapján készült.

Több *shared task* is indult az anafora- és a koreferenciafeloldás feladatával kapcsolatban, így ezek annotációs sémáit is érdemes megvizsgálni. A CoNLL-2012 ([Pradhan et al., 2012](#)) *shared task* annotációs sémája a koreferenciakapcsolatok két típusát különbözteti meg az *Identity* és az *Appositive* címkékkel. Az előbbi az anaforikus kapcsolatot, az utóbbi az attribúciós kapcsolatokat kódolja. A MUC-6⁶ és MUC-7 ([Hirschman és Chinchor, 1998](#)) *shared task*ok annotációs sémája nem alkalmaz különböző címkéket az egyes anafora- és koreferenciakapcsolat-típusokra. Ezek a sémák linkalapú koreferenciaannotációt tartalmaznak, ahol a linkek az entitások szövegbeli említéseit kötik össze. Ezeknek az annotációs sémáknak kifejezett célja, hogy magas annotátorok közötti egyetértést lehessen elérni velük. Ugyanakkor érdemes szem előtt tartani, hogy valójában sokkal több nyelvészeti

³<http://zil.ipipan.waw.pl/PolishCoreferenceCorpus>

⁴<https://brat.nlplab.org/>

⁵<http://cothec.nlp.ipipan.waw.pl/index.xhtmll#/>

⁶https://cs.nyu.edu/~grishman/COTask21.book_1.html

2.1. Pro-drop nyelvek koreferenciakorpuszai

tudás áll a rendelkezésünkre az anafora és a koreferencia jelenségeiről, ami különösen fontos lehet például az információkinyerési feladatok esetében.

2.1. Pro-drop nyelvek koreferenciakorpuszai

A dolgozat szempontjából a legérdekesebbek azok a koreferenciakorpuszok, amelyek pro-drop nyelvekre készültek, mert ezek a szövegben testesen meg nem jelenő, kiejtett (*dropped*) névmások szintén részt vesznek a szöveget átszövő anaforikus kapcsolatokban. Koreferenciakorpuszok léteznek pro-drop nyelvekre is, ilyen például az OntoNotes5.0 (Weischedel et al., 2013), ami arab és kínai szövegeket is tartalmaz, a japán NAIST Text corpus (Iida et al., 2007), a spanyol és katalán AnCora-CO (Recasens és Martí, 2010), a lengyel PCC (Ogrodniczuk et al., 2016) és a portugált is tartalmazó ParCorFull2.0 (Lapshinova-Koltunski et al., 2022). Az AnCora-CO szintaxisfái tartalmazzák a zéró alanyokat, és ezek a zéró elemek a koreferenciakapcsolatban is részt vesznek. A NAIST, OntoNotes5.0 és PCC korpuszok szintén tartalmazzák az adott nyelvekben létező zérónévmásokat.

A ZAC (Zero Anaphora Corpus) (Baptista et al., 2016) korpusz kifejezetten a zérónévmások antecedenskeresésére készült a brazil portugál nyelvre. 35 000 tokent tartalmaz különböző forrásokból származó szövegekből. Csak ez a nyelvi jelenség van annotálva benne, de ez nagyon részletesen. A zérónévmás számát és személyét is jelöli, valamint azt is, hogy anaforáról vagy kataforáról van-e szó. Különválasztja a mondatokon átívelő anaforikus kapcsolatokat is, valamint megadja az antecedens tokenjét is. Majdnem 1 500 zérónévmás szerepel a korpuszban, ami nagyon jól mutatja a téma fontosságát a pro-drop nyelvek esetében.

A ParCorFull2.0 (Lapshinova-Koltunski et al., 2022) egy párhuzamos korpusz, amiben az eredetileg csak angol és német nyelvű szövegeket tartalmazó ParCorFull korpuszt egészítették ki portugál és francia szövegekkel. A kihívást éppen az adta, hogy a portugált, egy pro-drop nyelvet kellett beilleszteni egy olyan korpuszba, ahol a készítőknak eredetileg erre a jelenségre nem kellett gondolnia. Éppen ezért a zéró névmások antecedense az ige mellett van megjelölve, ami jó megoldásnak tűnik, figyelembe véve, hogy az ige ragozása mutatja meg a zérónévmás jellemzőit. Ha ezt a megoldást alkalmazzuk a magyar nyelvre, akkor nem szabad megfeledkezni róla, hogy a magyar esetében egyszerre akár két zérónévmás is tartozhat egy igehez (alany és tárgy).

Az is egy lehetséges megoldás, ha a zérónévmások egyáltalán nem jelennek meg a korpuszban. Várhatóan a koreferenciafeloldó programok bemenetében sincsenek jelen testetlen elemek, és a kimenetben sem biztos, hogy meg kell jelenniük. Ezért ha például tanítóanyagot készítünk egy koreferen-

2.2. Magyar koreferenciakorpuszok

ciafeloldó számára, akkor nem feltétlenül van szükségünk a zéró elemek megjelenésére a tanítókorpuszban. Ugyanakkor az információkinyerési feladatok számára mindenképpen hasznos, ha minél gazdagabb nyelvészeti annotáció áll a rendelkezésünkre, ide értve a zéró elemeket (pl. zéró létigét, elliptált igéket, zérónévmásokat). A legjobb megoldás, ha a korpusz tartalmazza a zéró elemeket, de nélkülük is használható.

2.2. Magyar koreferenciakorpuszok

A SzegedKoref (Vincze et al., 2015, 2018) a Szeged Korpusz (Csendes et al., 2005) egy részét felhasználva készült, újsághíreket és iskolai fogalmazásokat láttak el koreferenciaannotációval. A legutóbbi publikáció alapján a SzegedKoref 400 szöveget, 4 021 mondatot és 55 763 tokent tartalmaz. A szövegekben 2 456 anaforikus láncot⁷ jelölték meg.

A SzegedKoref annotációs rétegei kézzel ellenőrzött minőségűek, így nemcsak az anafora- és koreferenciaannotáció hasznosítható belőle, hanem a többi nyelvi elemzés is. A SzegedKoref az MSD morfológiai kódkészlet⁸ (Erjavec, 2004) egy jegy-érték pár formátumban megfogalmazott verzióját⁹ használja morfológiai címkekészletként. Emellett a SzegedKoref összetevős mondatelemzést is tartalmaz.

A SzegedKoref az alábbi kapcsolattípusokat különbözteti meg: névmási, főnévi, határozói, igei és képzett. A főnévi osztályt az alábbi alkategóriákra osztották tovább: ismétlés, szinonima, hipernima, holonima, epitheton és appozíció. A SzegedKoref engedélykérés után kutatási és oktatási célokra felhasználható.

A magyar nyelvű koreferenciakorpuszok között meg kell említeni Miháltz et al. (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.

⁷Az anaforikus láncok magukban foglalják a névmási anaforikus kapcsolatokat és a koreferenciaviszonyokat is.

⁸<http://nl.ijs.si/ME/Vault/V3/msd/msd.pdf>

⁹https://github.com/dlt-rilmta/panmorph/blob/master/panmorph_conll.pdf

3. fejezet

Egy magyar koreferenciakorpusz: KorKor

Az ebben a fejezetben bemutatott munka célja egy gold standard minőségű korpusz létrehozása volt, amely anafora- és koreferenciaannotációt tartalmaz. A tervezési fázis kimenete egy pilotkorpusz, a KorKor, amely dokumentációival együtt elérhető a GitHubon¹. A pilotkorpusz mellett a teljes munkafolyamatot és az építéshez használt összes eszközt is elérhetővé tettem. A jelenlegi fejezet a munkafolyamatot és a korpuszépítés lépéseit részletesen ismertető (Vadász, 2020) és (Vadász, 2022) alapján készült.

A korpusz tervezésekor három fontos kritériumot fogalmaztam meg, amelyeknek a KorKor korpusznak meg kell felelnie:

- jó minőség
- elérhetőség
- reprodukálhatóság

Az anafora- és koreferenciaannotáció mellett a többi nyelvi annotáció (szófaj, morfoszintaktikai jegyek, függőségi elemzés) esetében is a legjobb minőségre törekedtem, a korpusz minden elemzési rétegét kézi annotációval készítettem el. Ennek érdekében a tokenizálástól kezdve egészen a függő-

¹https://github.com/vadno/korkor_pilot

3.1. Magyar előzmény: a SzegedKoref

ségi mondatelemzésig az automatikus annotációt kézi ellenőrzés² követte. Bizonyos testetlen nyelvi elemek jelölését kifejezetten fontosnak tartottam, mert ezek az elemek fontos szerepet játszanak a névmási anaforikus viszonyokban is.

Egy további fontos alapelv volt a korpusz tervezésekor, hogy a végeredmény bárki számára minél egyszerűbben hozzáférhető legyen. Ennek érdekében csak olyan szövegeket használtam, amelyek szabadon felhasználhatóak.

A reprodukálhatóságot pedig az alapos dokumentáció biztosítja. A korpusz és az annak építésekor használt saját fejlesztésű eszközök a GitHub-on elérhetőek, a dokumentációk segítségével az egyes korpuszpépítési lépések reprodukálhatók, saját szövegekkel kiterjeszthetők.

A KorKor tervezésekor a lehetséges hasznosítási módokat is számba vettem. A korpusz szövegei és az egyes elemzési rétegek közvetlenül is hasznosíthatók, a kézi annotálásból fakadóan alkalmasak hasonló elemzéseket kibocsátó eszközök betanítására vagy kiértékelésére.

3.1. Magyar előzmény: a SzegedKoref

A KorKor tervezésekor a 2.2. fejezetben már említett, a jelenleg létező legnagyobb magyar koreferenciakorpusz, a SzegedKoref nyújtott inspirációt.

Mi szükség van a SzegedKoref mellett még egy magyar koreferenciakorpuszra? Ez a kérdés több irányból is megközelíthető. A kézzel annotált adat nagyon értékes erőforrás, és minél több van belőle, annál jobb. A cikkben ismertetett KorKor összes elemzési rétege – a SzegedKorefhez hasonlóan – kézzel ellenőrzött minőségű, így nemcsak 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 kódkészlet³ (Erjavec, 2004) egy jegy-érték (*feature-value*) párok formájában megfogalmazott verzióját⁴ használja morfológiai címkészletként, addig a KorKor morfológiai rétege emMorph⁵ (Novák et al., 2017) és UD-kompatibilis⁶ morfológiai címkéket tartalmaz. Egy másik különbség, hogy a SzegedKoref összetevős elemzést, míg a KorKor dependencia-

²Hálámat fejezem ki az annotátoraimnak, Bencze Norbertnek, Bognár Ivettnek, Fegyő Kingának és Fodor Grétának, akik a monoton feladatok elvégzése mellett friss ötleteikkel folyamatosan inspiráltak.

³<http://nl.ijs.si/ME/Vault/V3/msd/msd.pdf>

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

⁵https://e-magyar.hu/en/textmodules/emmorph_codelist

⁶https://github.com/dlt-rilmta/panmorph/blob/master/panmorph_ud.pdf

3.1. Magyar előzmény: a SzegedKoref

elemzést használ szintaktikai elemzésre. Végül, a KorKor korpusszal 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 KorKor anyagával. Ehhez csupán az eltérő formátumú koreferenciaannotációt kell egységes formára hozni.

Ugyanakkor fontos megemlíteni, hogy a KorKor tervezésekor bizonyos elméleti kérdésekben eltérő döntéseket hoztam a SzegedKorefhez képest. A SzegedKoref sémájával szemben a KorKorban az infinitívusz alanya is megjelenik zéró névmásként, hiszen ezek az alanyok is részt vehetnek anaforikus kapcsolatokban. Emellett a KorKorba beillesztettem a zéró létigéket és az elliptált igéket, szintén azért, mert ezeknek a zéró igéknek az alanya is szerepelhet anaforikus kapcsolatban. A KorKor címkekészletében különválasztottam az anaforikus kapcsolatokat a koreferenciaviszonyoktól, hogy megkülönböztessem egymástól a szimmetrikus és nem szimmetrikus kapcsolattípusokat. A SzegedKoref az alábbi kapcsolattípusokat különbözteti meg: névmási, főnévi, határozói, igei és képzett. A főnévi osztályt az alábbi alkategóriákra osztották tovább: ismétlés, szinonima, hipernima, holonima, epitheton és appozíció. Ezzel szemben a KorKor címkekészlete mindössze kétféle alkategóriát különböztet meg a főnévi kapcsolatok között elkülönítve a rész-egész kapcsolatokat a közös referenciájú elemek között fennálló viszonytól. Ennek a különbségnek az az oka, hogy a korpusz annotálásakor az annotátorok számára nehézséget okozott elkülöníteni az egyes koreferenciátípusokat, ezért inkább a koreferenciakapcsolatokra vonatkozó címkekészlet egyszerűsítése mellett döntöttem. A KorKor címkekészlete a névmási anafora esetében viszont részletgazdagabb, a névmás fajtájának megfelelő osztályozást tartalmaz: személyes, mutató, visszaható, kölcsönös, vonatkozó és birtokos névmásokat megkülönböztetve, továbbá külön címkét tartalmaz az általános alany számára, valamint a beszélő és a címzett számára. A SzegedKoref annotációs útmutatója kitér rá, hogy az általános alanyt nem kell annotálni a korpuszban. Ennek ellenére a KorKor építése során a szövegekben sokszor úgy találtam, hogy az általános alany is részt vesz anaforikus láncokban. Az 1. példában látható, ahogy a tagmondatok igéinek alanya általános alany, és közöttük anaforikus kapcsolat áll fenn.

- (1) A kép módosításával **kiemelték** a korszak két legnagyobb vezetőjét, Maót és Sztálint. A nem fontos embereket **törölték**, hogy **kiemeljék** a vezető személyét.

A beszélő és a címzett a SzegedKoref annotációs útmutatója alapján ugyanazt a névmási címkét kapja, mint a többi névmás is, a KorKor esetében ezeknek külön címkét határoztam meg. Ugyan a KorKor jelenleg nem tartalmaz dialógust, de a beszélő és a címzett elkülönítése lehetőséget ad rá, hogy párbeszédet tartalmazó szövegekkel is bővíthető legyen a korpusz. Végül pedig a KorKorban nem jelöltem a határozói, igei és képzett kapcsolatokat.

3.2. A korpusz főbb adatai

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

3.2. A korpusz főbb adatai

A KorKor az `emt sv` (Indig et al., 2019b) keretében használt `xt sv` formátumot használja. A korpusz 94 dokumentumot, 1 351 mondatot és 26 583 tokent tartalmaz, amelybe beleszámítanak az írásjelek és a zéró elemek is.

Jelenleg két forrásból gyűjtött szövegeket tartalmaz a korpusz, amelyeket az OPUS gyűjteményéből (Tiedemann, 2012) válogattam. A szövegek egy részét a magyar Wikipédiáról gyűjtöttem, másrészt a GlobalVoices hírportál⁷ magyar nyelvre lefordított hírei közül válogattam. A KorKor örökliezeknek a forrásoknak a nyílt hozzáférhetőségét.

A korpusz az építéséhez készített eszközökkel és dokumentációikkal, valamint az annotálási útmutatókkal együtt a már fent említett GitHub repozitóriumban érhető el.

3.3. A korpuszépítés folyamata

A korpusz tervezésekor a munkát egy feldolgozó láncként képzeltem el. Céлом volt, hogy minél több lépést automatizáljak és emberi munkát csak az eszközök kimenetének javításához használjak.

Mivel az `emt sv` egy szövegfeldolgozó eszközlánc, 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 alkalmaztam, amely ugyan idő- és munkaigényes feladat, viszont könnyebben kontrollálható. Az `emt sv` kimenetét két körben volt szükséges ellenőrizni és javítani, ezután a saját eszközeim (zérónévmás-beszűrő, anaforafeloldó) kimenetét is ellenőrizni kellett.

⁷<https://hu.globalvoices.org>

3.3. A korpuszépítés folyamata

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 (emToken, emMorph, emTag)
3. formátumátalakítás (**saját szkript**)
4. kézi ellenőrzés (Google Spreadsheets)
5. formátumátalakítás (**saját szkript**)
6. elemzés (emDep)
7. formátumátalakítás (**emCoNLL**)
8. kézi ellenőrzés (WebAnno)
9. zéró létigék és elliptált igék 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 munkafázisokat annotátorokkal közösen végeztem el, akik mindannyian nyelvész végzettségűek voltak. Az annotátorokkal munkaidő-nyilvántartást vezettünk, ahol nem csak azt rögzítettem, hogy mely fájlokkal végeztünk, hanem azt is, hogy az adott fájl ellenőrzésekor milyen nehézségekbe ütköztünk. Ezen kívül azt is követtem, hogy az egyes fájlok ellenőrzése – az egyes elemzési szinteken – mennyi időt vett igénybe, ezáltal a korpusz további bővítésének költségei is kalkulálhatóak. A 3.1. táblázat azt mutatja, hogy átlagosan hány percet vett igénybe egy dokumentum ellenőrzése a különböző elemzési szinteken. Összehasonlításképp, a SzegedKoref annotálása személyes közlés alapján 1 óra/dokumentum volt, a teljes korpusz (55 763 token, 4 021 mondat, 400 dokumentum, átlagosan 10 mondat vagy 140 token dokumentumonként) kb. 400 munkaórát vehetett igénybe.

	perc/dokumentum	munkaóra
morfológiai egyértelműsítés ellenőrzése	0:24:13	38
függőségi elemzés ellenőrzése	0:29:23	46
anaforák beillesztésének ellenőrzése	0:34:22	54
összesen		138

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

3.3. A korpuszépítés folyamata

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

Az egyes lépések között a fájlok formátuma többször is változik, ahol az egymást követő két lépés bemeneti fájlformátuma eltér. A folyamat legutolsó lépése a fájlok átalakítása `xtsv` formátumra.

Egy rövid kitérő után, ahol egy, a projekt során készült összevetőprogramot ismertetek, részletezem a korpusz építésének egyes lépéseit.

3.3.1. Az `xtsv` fájlok összevetése: `emDiff`

A folyamatok ellenőrzéshez egy összevető programot is készítettem. Az `emDiff`⁸ lehetővé teszi az eltérő tokenizálású szövegek összefésülését⁹ és az egyes oszlopok tartalmának összevetését.

Már a korpusz építésekor felmerült, hogy szükség van egy eszközre, ami megkönnyíti az egyes munkafolyamatok értékelését. A munka során hasznos lehet például összevetni az annotátorok által ellenőrzött fájlokat, megfigyelni a gépi elemzés és a kézi ellenőrzés közötti különbségeket és természetesen megmérni az annotátorok közötti egyetértést különböző mérőszámokkal. Mivel az építés során a legtöbb lépésben az `emtsv` keretrendszerében működő modulokat használtam, valamint a saját szkriptjeimet is igyekeztem beilleszteni ebbe a keretrendszerbe, ezért kézenfekvő volt egy olyan eszköz készítése, amely szintén jól illeszkedik az `emtsv` kereteibe. Az `emDiff` ennek megfelelően az `emtsv` be- és kimeneti formátumának megfelelő fájlokat fogadja bemenetként.

A felhasználó több üzemmód közül választhat (akár egyszerre többet is), majd a program felismeri, hogy milyen elemzési rétegeket tartalmaznak a bemenetül kapott fájlok. Az elemzési rétegek metszetén végzi el a felhasználó által kijelölt feladatokat. A három választható üzemmód az alábbi feladatok elvégzésére szolgál:

- egyszerű összevetésre (az eltérések vizualizációjára),
- az egyik szöveget gold standardként tekintve kiértékelési feladatokra, és
- annotátorok közötti egyetértés számolására.

⁸<https://github.com/vadno/emdiff>

⁹a Python `diff` csomagjával (<https://docs.python.org/3/library/difflib.html>)

3.3. A korpuszépítés folyamata

A program a bemenetként kapott fájlok elemzési rétegei alapján különbözőképpen lát neki a feladatoknak. Az elemzési rétegeket a fájlok fejlécéből nyeri, hiszen az oszlopnevek jelölik azokat. Erre azért van szükség, mert más módon és más mérőszámokkal kell kiértékelni egy címkézési feladatot, mint például a függőségi elemzést. A program azokat az elemzési rétegeket veszi figyelembe, amelyek mindkét fájlban szerepelnek. A felhasználónak tehát nem kell kijelölnie, hogy melyik elemzési rétegekkel szeretne dolgozni és azokkal milyen számításokat szeretne végezni, az egyes oszlopok tartalmán a nekik megfelelő mérőszámokkal automatikusan lezajlanak a megfelelő számítások. Az eredmények eléréséhez a felhasználónak csak meg kell keresnie a kimenetekben a megfelelő oszlopokhoz tartozó kimeneti fájlokat.

A program mindegyik feladatnak hasonló módon lát neki, elsőként elkészíti a két összehasonlítandó fájl *deltáját*. A delta mindkét fájl sorait tartalmazza, a sorok elején jelöli, hogy az adott sor melyik fájlban van jelen. Ha mindkettőben, akkor a sor elejére két space karaktert illeszt, ha csak az első fájlban, akkor egy '-' jelet, ha csak a másodikban, akkor egy '+' jelet. A későbbi feldolgozás során a további feladatok elvégzéséhez a program ezt a deltát fogja használni. A delta fájl elkészítéséhez az emDiff a Python difflib csomagját használja.

Az emDiff a kiértékelést az `emt sv` keretében modulként leggyakrabban használt alapvető nyelvfeldolgozó programok kimenetein képes. Kiértékeléskor a bemenetként kapott két fájl közül az elsőt tekinti a gold standardnak, ahhoz hasonlítva értékeli ki a másodikat. A 3.2. táblázat mutatja, hogy az egyes oszlopokhoz milyen kiértékelési mérőszámokat kapunk. A táblázat első oszlopában az egyes elemzési rétegek nevei, az `emt sv` kimenetében az oszlopnevek láthatók. A második oszlopban azok a mérőszámok, amelyeket az emDiff biztosít ezekhez az elemzési rétegekhez. A `lemma` és az `xpostag` oszlopok az `emTag` által eredményezett egyértelműsített tövet és `emMorph` címkét tartalmazzák, a `upostag` és a `feats` pedig a Universal Dependencies címkézési keretében megfogalmazott szófajcímkét és inflexiós jegyeket. Az `NP-BIO` és a `NER-BIO` szekvenciális címkézési feladatok eredményei, az előbbi a névszói frázisokat, utóbbi pedig a névelemeket jelöli a szövegben az `emChunk` és az `emNer` kimeneteként. Ezeknek a címkézési feladatoknak a kiértékelésekor az IOB-accuracy mérőszámot tokenenként, a pontosságot, fedést és F-mértéket chunkonként számoljuk. Az `id`, `head` és `deprel` oszlopokban a függőségi mondatelemzés eredménye szerepel. A függőségi mondatelemzést két mérőszámmal értékelhetjük ki: a LAS (*labeled attachment score*) a függőségi kapcsolatot és a címketípust is figyelembe veszi, míg az UAS (*unlabeled attachment score*) pusztán a kapcsolatot. Utóbbi esetben tehát a függőségi fában elhelyezkedő élek helyességét figyeli, míg az előbbinél a helyesen felépített függőségi fában a címketípusoknak is megfelelőnek kell lenniük. A `cons` oszlop az összetevős mondatelemzést, az `emCons` kimenetét tartalmazza.

3.3. A korpuszépítés folyamata

oszlopnév	mérőszám
lemma	accuracy
xpostag	accuracy, pontosság, fedés, F-mérték, confusion matrix
upostag	accuracy, pontosság, fedés, F-mérték, confusion matrix
feats	accuracy, pontosság, fedés, F-mérték, confusion matrix
NP-BIO	IOB-accuracy, pontosság, fedés, F-mérték
NER-BIO	IOB-accuracy, pontosság, fedés, F-mérték
id, head, deprel	LAS, UAS
cons	accuracy, pontosság, fedés, F-mérték, confusion matrix

3.2. táblázat. Az egyes oszlopokhoz és elemzésekhez tartozó kiértékelő metrikák.

Az emDiff segítségével az `emt sv` keretein belül leggyakrabban használt eszközök kimenetén kívül egy saját szkript, a KorKor korpusz építéskor használt zérónévmás-beszűrő, az emZero kimenetét is kiértékelhetjük, ehhez a `zeroeval` üzemmódot kell kiválasztani. Mivel a zérónévmás-beszűrő alkalmazásakor nem kerül plusz oszlop a fájlba, ezért ez külön funkcióként került a programba, amelyet a felhasználónak kell kiválasztania a futtatáskor. Az emZero kimenetét három mérőszámmal értékeli ki: pontosság, fedés és F-mérték.

Az annotátorok közötti egyetértés számolása esetén a két bemeneti fájl a két annotátor által annotált fájl. A címkézési feladatok esetén az egyetértést a legismertebb mérőszámok segítségével fejezi ki, *megfigyelt egyetértés*, S , π , κ , súlyozott κ és Krippendorff α . Ezeknek az értékeknek a számolásához az emDiff az `NLTK.METRICS.AGREEMENT` csomagot¹⁰ használja. A függőségi elemzés esetében másfajta mérőszámokkal fejezzük ki az annotátorok közötti egyetértés mértékét, amelyek a következők: Az UAA (unlabeled attachment agreement) csak a függőségi kapcsolatot veszi számításba, az LOA (label only agreement) csak a címkét, az LAA (labeled attachment agreement) pedig a kapcsolatot és a címkét is.

Az emDiff tehát nemcsak a több annotátor által annotált ugyanazon szövegek összevetésére alkalmas, hanem annotátorok közötti egyetértés számítására is, valamint az annotátorokkal közösen egyeztetett 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 `emt sv` moduljaként is futtatható.

¹⁰https://www.nltk.org/_modules/nltk/metrics/agreement.html

3.3. A korpuszépítés folyamata

3.3.2. Szöveggyűjtés

A 3.2. fejezetben említett forrásokból több mondatot tartalmazó szövegeket gyűjtöttem, hiszen az anafora- és koreferenciaviszonyok mondathatárokon is átívelnek. A szövegek hossza 4 és 25 mondat között, a mondatok hossza 5 és 79 token között van (az írásjeleket külön tokennek számolva). A dokumentumok átlaghossza 14 mondat, mediánja 15 mondat, a mondatok átlaghossza 19 token, mediánja 18 token.

Az összegyűjtött szövegeket `emt sv` elemzőeszköz számára megfelelően kellett előkészíteni. Bár a pilotkorpusz szövegei standard helyesírásúak voltak, szükségesnek tartottam minden szöveget átneézni. Az annotátorokkal együtt 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) tároltam el, szövegenként külön fájlban.

3.3.3. Morfológiai elemzés és egyértelműsítés az `emt sv` elemzővel

Az `emt sv` megfelelő moduljai (`emToken` (Mittelholcz, 2017), `emMorph` (Novák, 2014; Novák et al., 2016b; Novák, 2003) és `emTag` (Orosz és Novák, 2012, 2013)) segítségével történő elemzés kimenete egy négy oszlopot tartalmazó fájl, aminek a formátumát röviden már a 3.2. 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.

3.3.4. A morfológiai címkék kézi javítása

Az első kézi javítás során az annotátorokkal a tokenizálás és a morfológiai egyértelműsítés eredményét ellenőriztük. Bár a szövegeket elemzés előtt átolvastuk, hogy mindenképpen standard helyesírásúak legyenek, előfordult, hogy helyesírási vagy nyelvhelyességi hiba maradt a szövegekben. Ezeknek a hibáknak a javítása is ebben a körben történt.

A javítás alapötlete az volt, hogy az egyes tokenekhez a gépi egyértelműsítés eredményéül kapott címkék mellett a morfológiai elemző által kibocsátott összes lehetséges címkét is láthatóvá teszem az annotátorok számára. Ennek az oka az a feltételezés, hogy ha az egyértelműsítő hibázik is, a le-

3.3. A korpuszépítés folyamata

hetséges elemzések között szerepelhet a helyes címke. Így az annotátoroknak csak abban az esetben kell kézzel megadniuk egy morfológiai címkét, ha nemcsak az egyértelműsítő, hanem a morfológiai elemző is hibázott, így gyorsabban, és kevesebb hibalehetőséggel dolgozhatnak. Szerencsés, ha az annotátornak a kézi címkebevitelhez csak a legvégső esetben kell fordulnia, így csökkenthető a rosszulformált címkék elfordulásának valószínűsége. A rosszulformált címkék elkerülésére megoldást jelentene az is, ha az annotátor csak a lehetséges címkék listájából válogathatna – például egy legördülő listából – ahelyett, hogy kézzel építi fel a teljes címkét, azonban az emMorph címkék szerkezete ezt nem teszi lehetővé, hiszen több mint kétezerféle címkét lehet előállítani ebből a címkekészletből¹¹.

A javítást két annotátor segítségével hárman végeztük, akik mindketten nyelvészek, számítógépes nyelvészet szakirányt végeztek a munka tartama alatt. Az annotálási feladat során folyamatosan kommunikáltunk egymással, megbeszéltük a vitás eseteket és folyamatos visszajelzést kértem tőlük a feladat nehézségével, az útmutató pontosságával és követhetőségével, valamint az annotáló felülettel kapcsolatban.

A javításhoz egy megfelelő annotáló segédeszközre volt szükség. Fontos szempont volt, hogy az annotálás során megőrződjön a kimenetül kapott fájlformátum, hiszen a következő lépésben az `emt_sv` számára fogyaszthatónak kell lennie. A morfológiai elemzés címkekészletét is figyelembe kell venni az annotációs segédeszköz kiválasztásakor. Ezen kívül a javító felületnek könnyen használhatónak, ergonomikusnak kell lennie. A legegyszerűbb, ha az annotáló segédeszköz használatához nem szükséges az annotátor számítógépére telepíteni semmit, így olyan megoldást kerestem, amely böngészőből hozzáférhető. A verziókövetés, nyomonkövethetőség és kollaboratív munka is szerepelt a kívánt funkciók listáján.

A fenti kritériumoknak egyidejűleg egy annotáló segédeszköz sem felelt meg. A legegyszerűbb megoldásnak a Google Spreadsheets mutatkozott. A be- és kimeneti formátum megőrzése bizonyos elő- és utófeldolgozó lépések beillesztésével könnyen megoldható, hiszen a Spreadsheets-be egyszerű `t_sv`-ként importálható a bemenet, míg a kimenet `csv`-ként exportálható. A Spreadsheets felületével minden annotátor találkozott már, használata így nem tűnt bonyolultnak. A feltételes formázásoknak köszönhetően igyekeztem könnyen használható, ergonomikus felületet biztosítani az egyébként monoton, de nagy koncentrációt igénylő munkához. A használatához nem kell telepíteni semmit, böngészőből használható, kollaboratív és nyomonkövethető. A Google fájlkezelési protokollja biztosítja a verziókövetést is.

¹¹Az emMorph címkék felépítésével kapcsolatban lásd a 3.5. alfejezetet.

3.3. A korpuszépítés folyamata

Az annotátorok a saját nevükre szóló mappába kapták az előkészített (összegyűjtött, átnézett, `emtsv`-vel tokenizált, morfológiailag elemzett és egyértelműsített), valamint a Google Spreadsheets számára emészthető formátumra alakított szövegeket. Ehhez egy egyszerű szkript alakította át az `emtsv` kimenetét úgy, hogy az összes kinyerhető információt felhasználva előálljon egy áttekinthető és kezelhető dokumentum. A szövegeket egy feltételes formázásokkal előzetesen ellátott üres dokumentumba importáltam. A feltételes formázások célja, hogy vezessék az annotátor szemét a munka során, kiemeljék a potenciálisan javítandó elemeket, valamint visszacsatolást nyújtsanak a javítás folyamata során.

A dokumentumok felépítése a következő. Egy sorban egy token van, a mondatokat üres sor választja el egymástól. Minden új token sorát kék színnel emeltem ki. Minden tokenhez látható volt az összes lehetséges morfológiai elemzése, valamint ki volt emelve az az elemzés, amelyet az egyértelműsítő választott. Az egyértelműsítő által kiválasztott elemzés mellett egy 'X' volt látható. A feladat az volt, hogy az annotátor eldöntse, hogy megfelelő elemzést választott-e ki az egyértelműsítő. Ha nem, akkor az 'X' átkerült abba a sorba, amelyikben a helyes elemzés szerepel. Ha a morfológiai elemzések közül egyik sem volt helyes, akkor kézzel is meg lehetett adni a megfelelő elemzést. A kézi megadáshoz az annotálási útmutató nyújtott részletes segítséget, hogy csak jólformált címkék kerülhessenek a kimenetbe.

A felület továbbá lehetőséget nyújtott egyéb hibák javítására is. Ha az annotátor elírást vagy hibás szóalakot talált, akkor a tokent is ki lehetett javítani. Ha a tövesítés nem sikerült, akkor a tövet is meg lehetett adni kézzel. A tokenizálási hibák javítására kézzel beírható parancsokat határoztam meg, amelyeket aztán az exportált `csv` feldolgozásakor automatikusan értelmezett egy szkript és azoknak megfelelően módosította a tokenizálást (sorról törölt vagy sort szűrt be az annotátor által megadott tartalommal).

Az exportált `csv` feldolgozásakor a tokenizálási javításokra vonatkozó parancsok értelmezése mellett az `emtsv` formátumára történő átalakítás is megtörtént. A kézi javítás után a szöveg tehát pontosan ugyanúgy nézett ki, mint javítás előtt, különbség a kijavított mezőkben van csupán. Így a szöveg készen állt arra, hogy továbbhaladjon az elemzési lépések során.

A kézi javítást egy közösen szerkeszthető Google táblázat segítségével követtem nyomon, amelyben az annotátorok az egyes fájlok mellett jelölték, hogy mennyi időt vett igénybe a javítás és jelezheték azt is, ha valami nehézség merült fel a javítás során. A munkaidő nyilvántartása nagy segítséget jelent a korpusz bővítése esetén, hiszen könnyen kiszámolható, hogy átlagosan mennyi idő alatt készült el egy-egy fájl javítása, így tervezhető a bővítés költsége is.

3.3. A korpuszépítés folyamata

A szövegek egy részét mindhárman ellenőriztük, 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, az UD¹² alapján dolgozik. Hogy az ebből fakadó különbségeket ne vegyük figyelembe az annotátorok közötti egyetértés számolásakor, már az emMorph-ról erre az inflexiós jegyeket tartalmazó készletre konvertált¹³ címkéket használtuk. A 4 315 tokennyi, mindhárom annotátor által ellenőrzött szövegre kapott eredmény Krippendorff alfában (Artstein és Poesio, 2008) kifejezve: 0,976. Ez az eredmény a Landis-Koch skálán (Landis és Koch, 1977) értelmezve majdnem tökéletes egyetértési mértéknek tekinthető.

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

Ezt a javítási módszert a pontos dokumentáció miatt más projektekben is lehet használni. Az NYTK-NerKor (Simon és Vadász, 2021) korpusz építéskor egyrészt felhasználtuk a KorKor korpusz anyagát, másrészt a további szövegek morfológiai címkéinek kézi ellenőrzéséhez alkalmaztuk az itt bemutatott ellenőrzési módszert és eszközöket. Az NYTK-NerKor építéséről a 4. fejezet szól.

3.3.5. Az egyértelműsítő kiértékelése

A KorKor kézzel javított elemzési rétegei lehetővé teszik, hogy megvizsgáljuk, hogy a korpuszépítéshez használt egyes eszközök milyen minőségű elemzést biztosítottak, így nemcsak a KorKor építéséhez újonnan készült előelemző eszközök, hanem az emt_{sv}-ben használt modulok teljesítménye is kimérhető, ezért ebben a fejezetben néhány emt_{sv} modul kiértékelésére használtam a korpuszt. A projekt részletei egy GitHub repozitóriumban¹⁴ is elérhetők.

Az emTag (Orosz és Novák, 2012, 2013) teljesítményét a 3.3.1. fejezetben ismertetett emDiff segítségével vizsgáltam meg a KorKor korpusz anyagán.

A kiértékelendő anyag tehát az emTag kimenete, az elemzést az emt_{sv} segítségével végeztem 2019 februárjában. Ezt az anyagot vettem össze a KorKor korpuszba épített, az annotátorok segítségével kézzel ellenőrzött és javított címkéket tartalmazó anyaggal (amit a 3.3.4. alfejezet már ismerte-

¹²<https://universaldependencies.org/>

¹³A konverzióról részletesebben a 3.5. fejezet ír.

¹⁴https://github.com/vadno/korkor_kiserletek

3.3. A korpuszépítés folyamata

tett). Az egyértelműsített fő és az egyértelműsített morfológiai címke mezők tartalmát is kiértékeltem. A vizsgálathoz használt anyag méretét a 3.3. táblázat tartalmazza.

	darabszám
fájl	121
mondat	1 720
token	36 492

3.3. táblázat. Az emTag kiértékeléséhez használt anyag mérete a gold standard adaton számolva. A tokenszámba az írásjeleket is beleszámítottam.

Az eredményeket a 3.4. táblázat tartalmazza. Az **összesített** eredmény a teljes anyagra számolt pontosságot (*accuracy*) mutatja, az **átlag** pedig a fájlonként elért pontosság átlaga a teljes korpuszra nézve.

	átlag	összesített
tő	98,23%	98,15%
morfológiai címke	95,48%	95,40%

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

Az eredmény alapján elmondható, hogy kevés esetben kellett kézzel kijavítani a címkéket. Az emTag által kibocsátott és a kézzel javított fájlok összevetése, valamint az annotátorok visszajelzései alapján vannak tipikus hibák, amelyek egyértelműen az emTag tanítóanyagának hibáiból erednek. Mivel a 4. fejezetben ismertetett NYTK-NerKor projektben egy jóval nagyobb adathalmazt annotáltunk, amelynek az elkészítéséhez pontosan ugyanazt a módszert használtuk, mint a KorKor esetében, valamint a megfigyeléseink megegyeztek az emTag kiértékelésekor tapasztalt hibákkal, ezért az emTag által vétett tipikus hibákért lásd a 4.6. alfejezetet.

3.3.6. Függőségi elemzés az emt sv elemzővel

A címkék kézi javítása után a szövegek szintaktikai elemzését az emt sv függőségi elemző modulja végezte el. A függőségi elemzés előtt az emMorph címkét át kellett alakítani a függőségi elemző modul számára emészhető UD-kompatibilis morfológiai címkére, amely megegyezik a Szeged Depen-

3.3. A korpuszépítés folyamata

dency Treebank (Vincze et al., 2010) és az `emt_sv` függőségi elemzője¹⁵ (Zsibrita et al., 2013), az `emDep` címkekészletével¹⁶. Ezt a konverziót is az `emt_sv` egy modulja, az általam készített `emmorph2ud`¹⁷ végezte el. Az UD-re való konvertáláskor a derivációra vonatkozó egyes információk elvesznek. A morfológiai címkekészletekről és a konverzióról részletesebben a 3.5. fejezet ír. 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 `emt_sv` formátumának megfelelően visszaalakított szövegeket az `emt_sv` függőségi elemzőjével elemeztem.

3.3.7. A függőségi elemzés kézi javítása

Az annotációt segítő eszközök célja, hogy minél ergonomikusabbá tegyék a munkát az annotátorok számára, legyen szó bármilyen szövegannotálási feladatról. A legfrissebb ilyen eszközöknek rengeteg elvárásnak meg kell felelniük, túl azon, hogy felületet biztosítanak a megfelelő címkézési feladathoz. A rugalmas be- és kimeneti formátumok lehetővé teszik, a legkülönfélébb elő- és utófeldolgozó eszközök használhatóak legyenek a szövegekre, de gyakran beépített feldolgozóeszközöket is tartalmaznak. Az online, valós idejű, kollaboratív felület még rugalmasabbá teszi a munkát, valamint lehetővé teszi az annotátorok teljesítményének monitorozását a hatékonyság növelése érdekében. Ezek az eszközök gyakran olyan fontos statisztikák előállítására is alkalmasak, mint például az annotátorok közötti egyetértés. Nem utolsó sorban egyre letisztultabb és átláthatóbb felületek állnak rendelkezésre. A legújabb eszközök működtetéséhez általában elég egy böngésző és internetkapcsolat, nem szükséges lokálisan telepíteni semmilyen programot a használatukhoz. Ilyen online használható annotáló eszköz például a `LightTag`¹⁸, a `Daturks`¹⁹ és a `tagtog`²⁰. Az egyik legelterjedtebb eszköz a `brat`²¹, amelyet több koreferenciakorpusz elkészítésénél is alkalmaztak.

¹⁵<http://e-magyar.hu/hu/textmodules/emdep>

¹⁶A címkekészletet lásd (Vincze et al., 2010).

¹⁷<https://github.com/vadno/emmorph2ud>

¹⁸<https://www.lighttag.io/>

¹⁹<https://daturks.com/index.php>

²⁰<https://www.tagtog.net/>

²¹<http://brat.nlplab.org/index.html>

3.3. A korpuszpépítés folyamata

A függőségi elemzés kézi javításához először meg kellett találni a megfelelő annotációs segéd-eszközt. A UD honlapja²² több olyan eszközt sorol, amelyek az `emt sv` által is kibocsátott CoNLL-U formátumot használják bemeneti formalizmusként, így ezek közül válogathattam.

A bemeneti formalizmus mellett több szempontnak is meg kellett felelnie a kiválasztott annotációs eszköznek. A függőségi elemzést legegyszerűbben egy drag-and-drop eszközzel lehet annotálni. Ezen kívül nagyon fontos volt, hogy ne csak plusz annotációs réteg felvitelére legyen alkalmas, hanem már meglévő annotáció kézi javítására is.

A javításhoz a WebAnno²³ általános célú, webalapú eszközt (Eckart de Castilho et al., 2016) választottam, hiszen a legtöbb fontos szempontnak megfelelt. 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 az annotátorok által módosított dokumentumok összevetése, a munka egyszerű nyomonkövetése és az annotátorok közötti egyetértés különböző mérőszámok alapján történő automatikus kiszámolása. Az eszköz rugalmasnak mondható, hiszen saját elemzési rétegek címkékészleteit is megfogalmazhatjuk. A WebAnno egy szerveren fut, az annotátorok pedig a megszokott böngészőjükön keresztül használhatják. A függőségi elemzés után az `emt sv` moduljaként működő konverterrel²⁴ alakítottam át a kimenetet a WebAnno számára emészthető CoNLL-U formátumra. A javítást két másik annotátor segítségével hárman végeztük.

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 találoztunk ilyen hibákkal. Sajnos a WebAnno felületén token törlésére vagy beszúrására nincs mód, így ezt a problémát egy utófeldolgozó lépésben kellett kezelni.

A korpuszban használt címkékészlet egyetlen címkében eltér az előelemzéshez használt függőségi elemzőétől. Az `emDep` elemző nem tart fenn külön címkét a birtok és birtokosa közt fennálló kapcsolat ábrázolására, azonban a birtokos kapcsolatokat a zérónévmások miatt fontosnak tartottam megjeleníteni. A birtokos és birtoka közti kapcsolat ábrázolására az `ATT` címkét használja, ami azonban más típusú kapcsolatokban is megjelenik (pl. főnév és módosítója közt). A korpuszban ezzel szemben minden birtokos kapcsolatot az eredeti címkékészletben nem szereplő `POSS` címkével lát-

²²<https://universaldependencies.org/tools.html>

²³<https://webanno.github.io/webanno>

²⁴<https://github.com/vadno/emconll>

3.3. A korpuszépítés folyamata

tuk el. Fontos megjegyezni, hogy a korpusz építésekor használt további feldolgozó eszközök is erre a bővített függőségi címkekészletre vannak felkészítve²⁵.

3.3.8. A függőségi elemző kiértékelése

A függőségi elemzés kézi javítása után lehetőségem volt a függőségi elemző teljesítményét is megvizsgálni. A kiértékeléshez²⁶ a KorKor korpusz egy részét használtam, mert a korpusz építése során ezeken a fájlokon nem történt tokenzám-módosító javítás, így ugyanolyan tokenzámú fájlokat és mondatokat lehetett összehasonlítani. A fájlok nem a végleges KorKor korpuszból vannak, hanem az annak építésekor keletkező köztes állapotú fájlok. Ennek az az oka, hogy a KorKor korpusz tartalmaz zéró elemeket (zéró létigéket és névmásokat), ezzel szemben a függőségi elemző nem produkál ilyeneket.

Kétféleképpen értékeltem ki az emDep teljesítményét. Az első kísérletben a dependenciaelemzőt azokon a fájlokon alkalmaztam, amelyekben az egyértelműsített morfológiai címke már kézzel javítva lett. A második kísérletben ugyanezeknek a fájloknak a kézi javítás előtti állapotát használtam. A két kísérlet eredményét tekintve következtethetünk arra, hogy hogyan javul a dependenciaelemzés minősége, ha a bemenetül szolgáló elemzési réteg minősége is jobb.

Az első kísérletben a kiértékelendő anyag tehát az emDep kimenete a javított morfológiai egyértelműsítés után, az elemzést az `emt_sv` segítségével végeztem 2019 augusztusában. A második kísérletben a kiértékelendő anyag az emDep kimenete a még nem javított morfológiai egyértelműsítés után, az elemzést az `emt_sv` segítségével végeztem 2020 augusztusában. A gold standard anyag ez emDep kézzel javított kimenete a WebAnno eszköz segítségével, amely a KorKor korpusz építése során jött létre (3.3.7). A vizsgálatához használt anyag méretét a 3.5. táblázat tartalmazza.

Az összevetéshez az emDiff kiértékelő modulját használtam, ami képes függőségi elemzés kiértékelésére is. Ehhez elsőként konvertálni kellett a WebAnno kimeneteként kapott CoNLL-U fájlokat `xt_sv` formátumra. A konvertáláshoz az általam készített, az `emt_sv` moduljaként is használható `conllu2xtsv` konvertert használtam.

Az eredményeket a 3.6 táblázat tartalmazza, a függőségi elemzés kiértékelésére használatos mérőszámokban (LAS, UAS) kifejezve. Az LAS (*labeled attachment score*) a függőségi elemzéshez tartozó

²⁵Ez alól a 7.2. fejezetben ismertetett, az `emt_sv` elemzőbe modulként beépített `emZero` kivétel, hiszen azt várjuk, hogy a felhasználó által elemezni kívánt bemenet az emDep eredeti sémájával és címkekészletével készült.

²⁶https://github.com/vadno/korkor_kiserletek

3.3. A korpuszépítés folyamata

	darabszám
fájl	96
mondat	1 530
token	29 823

3.5. táblázat. Az emDep kiértékeléséhez használt anyag mérete a gold standard adaton számolva. A tokenszámba az írásjeleket is beleszámítottam.

két oszlop tartalmát vizsgálja: az anyacsomópont index-számát és a vele fennálló kapcsolat típusát, magát a címkét. Ezzel szemben az UAS (*unlabeled attachment score*) csak a függőségi viszonyokat veszi figyelembe, a címkét nem. Ebben az esetben jó találatnak számít, ha a megfelelő anyacsomóponttól függ egy szó, mindegy, milyen kapcsolattípussal. Az UAS és LAS mérőszámok kiszámíthatók az egyes mondatokra, az egyes fájlokra és a vizsgálandó anyag egészére is. A kiértékelés során a teljes korpuszon egyben és fájlanként is kiszámítottam az UAS és LAS értékeket, az utóbbi esetben átlagszámítással következtettem a teljes anyag minőségére.

	első kísérlet		második kísérlet	
	átlag	összesített	átlag	összesített
LAS	0,83	0,83	0,77	0,77
UAS	0,87	0,88	0,89	0,89

3.6. táblázat. Az emDep teljesítménye LAS és UAS értékekben kifejezve.

A KorKor korpusz néhány esetben eltér az emDep által használt címkékészletől. Ezeket az eltéréseket kivételként kellett kezelni a kiértékelés során. Az emDep ATT címkét ad a birtokos és birtoka közötti függőségi élre, míg a KorKor korpuszban ez a kapcsolat meg van különböztetve a többi ATT címkével jelölt kapcsolattól és POSS címkével jelöli. Ezt a különbséget nem tekintem hibának és a kezelése is egyszerű. A kiértékeléskor egyszerűen minden POSS címkét ATT címkéként kell tekinteni.

Sajnos nem találtam arra vonatkozó információt, hogy a magyarul (Zsibrita et al., 2013) publikálásakor mért eredményt javított morfológiai címkéket használtak-e, viszont az eredményeken látható, hogy a függőségi elemzés minősége a KorKor-on mérve alacsonyabb, mint a magyarul (Zsibrita et al., 2013) publikálásakor elért teljesítmény (LAS: 91,42%, UAS: 93,22%), még az első kísérletben is, amihez a kézzel javított morfológiai címkéket használtam a függőségi elemzés előállításakor. Ennek az egyik lehetséges oka a szövegek eltérő műfaja lehet, igaz, a KorKor anyaga nem kellene, hogy nagy kihívást jelentsen, hiszen sztenderd, jó helyesírású, szerkesztett szövegekről van szó.

3.3. A korpuszépítés folyamata

Az emDep által vétett hibákat a korpusz építésekor az annotátorokkal javítanunk kellett. Az annotátorok által megfigyelt tipikus hibák pontosabb képet adhatnak róla, hogy mely pontokon tévedett a függőségi elemző. Az egyik ilyen tipikus hiba volt, hogy az alárendelő összetett mondatoknál a főmondat igéjének alanya az alárendelt mondat igéjéhez volt hozzácsatolva tévesen.

A KorKor korpuszon mért gyengébb eredmény egyik lehetséges oka, hogy bizonyos esetekben nem a függőségi elemző vétett hibát, hanem a KorKor korpusz függőségi annotációs sémája tért el túlságosan az emTeg tanítókorpuszának sémájától. A KorKor annotálásakor feldolgoztam a Szeged Treebank annotálási útmutatóját, azonban a függőségi mondatelemzés egy nagyon komplex rendszer, amelyhez a Szeged Treebankban – és így a KorKorban is – egy limitált címkekészlet párosul. Gyakran okozott nehézséget például annotálásakor a vonzatok és a szabad határozók elkülönítése. Kemény diónak bizonyultak a predikatív névszók, az összetett mondatok, valamint az elliptikus szerkezetek is. A részletes annotációs útmutató²⁷ mellé éppen ezért készült egy külön dokumentum²⁸ a nehezebb esetekkel. Mivel a Szeged Treebank annotálási útmutatója kevésbé részletes, nem lehetek benne biztos, hogy minden jelenséget pontosan ugyanúgy kezeltem a KorKor sémájának kidolgozásakor, a magas annotátorok közötti egyetértés ellenére is előforduló annotálási hibákról nem is beszélve.

3.3.9. A zéró létigék beillesztése

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. A 3.1. példában egy olyan mondat látható, ahol a függőségi fába két zéró létigét is be kellett illeszteni.

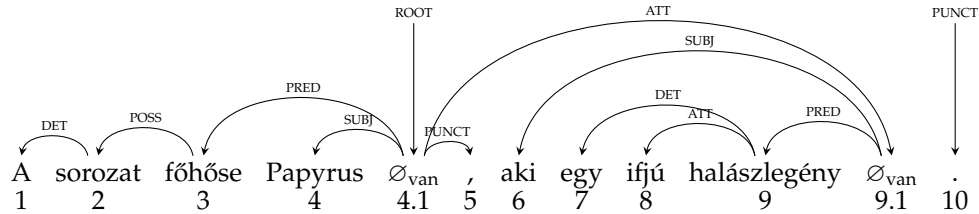
Az igei ellipsziseket is jelöltük a korpuszban, hiszen gyakran találoztunk olyan tagmondatokkal, amelyekben az elliptált ige hiánya miatt nem lehetett megfelelő anyacsomóponthoz kötni az egyes bővítményeket. A zéró létigékhez hasonlóan kézzel illesztettük a mondatfába az elliptált finit igéket. Az elliptált ige a zéró létigéhez hasonló, kombinált indexet kapott. A 3.2. példában egy olyan mondat látható, ahol a függőségi fába egy elliptált igét kellett beilleszteni.

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

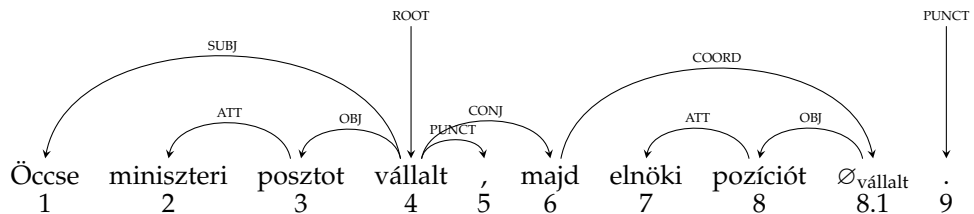
²⁷https://github.com/vadno/korkor_pilot/blob/master/guidelines/emdep_checker_guide.pdf

²⁸https://github.com/vadno/korkor_pilot/blob/master/guidelines/emdep_checker_faq.pdf

3.3. A korpuszépítés folyamata



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



3.2. 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.

3.3.10. A zérónévmás-beszúró

A zérónévmásokat egy saját szkript²⁹ 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

²⁹A szkript működését és kiértékelését a 7. fejezet ismerteti részletesebben.

3.3. A korpuszpépítés folyamata

A zérónévmások beillesztése után a mondatfában plusz ágak jelennek meg. A zéró elemek is saját ID-t kapnak, a t_{sv} -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 zérónévmás-beszűrő eredményét kézzel ellenőrizni és javítani kellett, ezt a feladatot azonban a későbbi lépésekben bekerülő anaforikus kapcsolatok ellenőrzésével és javításával egyszerre végeztük el, lásd 3.3.12. alfejezetben. Az annotátoroknak a javításkor különösen figyelniük kellett azokra a segédigékre (*kell, fog*), amelyeknél a szabályalapú zérónévmás-beszűrő automatikusan beillesztett egy zéró alanyt. Ezekben az esetekben törölniük kellett a beillesztett névmásokat.

3.3.11. A névmási anaforikus kapcsolatok beillesztése

A következő lépésben a névmási anaforikus kapcsolatokat is egy szabályalapú szkript szűri be, amely csak a személyes névmások előzményét keresi meg, a többi típust kézzel kellett beilleszteni. A személyes névmások előzménykeresésének egyszerű algoritmusát Pléh és Radics (1976) alapján dolgoztam ki, a program működését részletesebben a 8.2. fejezet ismerteti. A szkript kimenetét kézzel ellenőrizni és javítani kellett, amit a beillesztett zérónévmások ellenőrzésével és javításával együtt végeztünk el.

3.3.12. Kézi ellenőrzés és koreferenciaannotálás

Az automatikusan beillesztett zérónévmások és névmási anaforikus kapcsolatok ellenőrzését, valamint a koreferenciaannotálást három másik annotátor segítségével végeztük.

Számos annotációs eszköz található, amelyek segítségével lehet anafora- és koreferenciakapcsolatokat annotálni a szövegekben (pl. WebAnno, brat³⁰ (Stenetorp et al., 2012), TrEd³¹ stb.). Vannak olyan eszközök is, amelyek az annotáció javítására használhatók és például CoNLL-U formátumban képesek feldolgozni az adatot. Legjobb tudomásom szerint olyan eszköz azonban nem áll rendelkezésre, amely minden fontos kritériumunknak megfelelt volna.

³⁰<http://brat.nlplab.org>

³¹<http://ufal.mff.cuni.cz/tred>

3.3. A korpuszépítés folyamata

Az automatikusan beillesztett zérónévmasok 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, amelyekben ezek szerepelnek. Ilyen az általános alany szerepben álló zéró névmás, amelynek referenciája nehezen megragadható (**vastag betűvel** kiemelve az általános referenciájú alannyal rendelkező igéket (lásd a 2a. és a 2b. példákat).

- (2) 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 (3. példa, ahol **vastag betűvel** kiemeltük azokat az igéket, amelyeknek alánya 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 esetleges bővítése miatt gondolni kellett rá.

- (3) A születésnap 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.

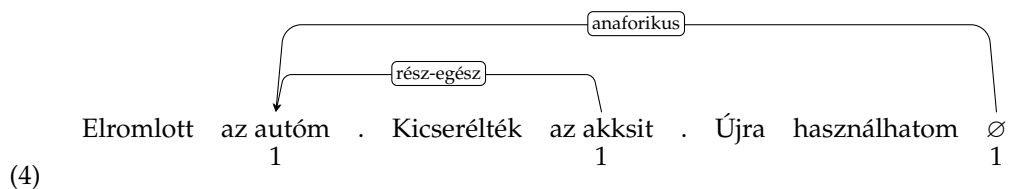
Új címkéket vezettem be ezekre az esetre: az *arb* az általános alany, az *addr* a címzett, a *speak* a beszélő/író referenciáját jelöli. Azzal, hogy bevezetjük a szöveg szereplői közé a beszélőt és a címzettet, jelölni tudjuk, ha a szövegben szereplő névmások ezen szereplők valamelyikével koreferens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

3.3. A korpuszépítés folyamata

Az elkészült korpuszban 274 olyan kapcsolat van, amelyben általános alany vesz részt, viszont a beszélő és a címzett sokkal ritkábban fordult elő. Ennek az oka az, hogy a Wikipédia szövegek és a hírek ritkán fordulnak az olvasóhoz, ritkán szólítják meg őt.

Az egyes koreferenciakorpuszokban, így a SzegedKoref annotációjában is különböző típusú koreferenciakapcsolatokat (pl. ismétlés, variáció, szinonima, hipernima, hiponima és holonima stb.) jelölnek. Az annotálás tervezése során és a szöveget megfigyelve azonban számos nehézségbe ütköztünk ezekkel a típusokkal kapcsolatban, gyakran nehéz volt különbséget tenni közöttük. Emellett viszont gyakran találoztunk rész-egész kapcsolatokkal a szövegben. 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. Ezek miatt az elágazások és kitérők miatt tartottam alkalmasabbnak a linkalapú annotációt az entitásalapú ábrázolással szemben. A 4. példán látható, hogy az entitásalapú annotáció alkalmazásával nem lehet megfelelően ábrázolni azokat az eseteket, amikor például egy rész-egész kapcsolatban részt vevő elem antecedense egy anaforának.



A példamondat fölött a linkalapú annotáció szerepel, alatta pedig az entitásalapú. A linkalapú annotációban esetben jól látható, hogy anaforikus kapcsolat az első mondat alánya és a harmadik mondat zéró tárgya között áll fenn, míg ha az entitásalapú annotáció alapján közös klaszterbe soroljuk az első mondat alanyát és a második mondat tárgyát, akkor nem maradt lehetőségünk arra, hogy pontosan ábrázoljuk az anafora és az antecedense közti kapcsolatot. A kétféle jelölésmód között az egyik irányból van átjárás: a linkalapú annotáció könnyen átalakítható entitásalapúvá, a másik irányból viszont nem lehet automatikusan konvertálni.

3.3. A korpuszépítés folyamata

A 3.7. 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

3.7. táblázat. Az anaforikus és koreferenciakapcsolatok előfordulása a KorKor korpuszban.

3.3.13. Formátumok

A korpusz többféle formátumban is elérhető. A KorKor.*xtsv* *xtsv* formátumot használ, így a korpusz ennek köszönhetően alkalmas további feldolgozásra az *emtsv* eszközeivel.

A 3.3. ábrán látható egy *xtsv* formátumú korpuszfájl. A fájlok első sorában az *xtsv* formátumnak megfelelően az elemzéseket tartalmazó oszlopok nevei szerepelnek. A példában látható néhány beillesztett zérónévmás is, amelyek sorszámozása úgy épül fel, hogy az eredetileg kiosztott sorszámozás is megmaradt.

A KorKor.*conllup* CoNLL-U Plus fájlformátumot használ. A KorKor.*conllup* fájlokat a KorKor.*xtsv* fájlokból konvertáltam. Noha ez a két fájlformátum átjárható, nemcsak egy egyszerű formátumkonverzió történt, a fájlok tartalmukban is különböznek. A zéró elemek (igék, névmások) a KorKor.*conllup* fájlokban nem szerepelnek önálló tokenként, tehát a függőségi fák és az anaforikus kapcsolatok tekintetében is találunk különbségeket. A zérónévmások is másként vannak annotálva a KorKor.*conllup* fájlokban: a zéró alanyok, tárgyak és birtokosok egy külön oszlopban, az ige vagy

3.4. Nehézségek

```

# global.columns = ID FORM LEMMA XPOS UPOS FEATS DEPREL HEAD COREFHEAD COREFTYPE ZERO_SUBJ ZERO_OBJ ZERO_POSS
# sent_id = globv.1.conllup_1
1 Három három [/N][Nom] NUM Case=Nom|Number=Sing|NumType=Card ATT 2 - - - - -
2 hónap hónap [/N][Nom] NOUN Case=Nom|Number=Sing SUBJ 3 - - - - -
3 telt telik [/V][Pst.NDef.3Sg] VERB Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act ROOT 0 - - - - -
4 el el [/Prev] PREVERB 3 - - - - -
5 az az [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 7 - - - - -
6 újságíró újságíró [/N][Nom] NOUN Case=Nom|Number=Sing ATT 7 - - - - -
7 házaspár házaspár [/N][Nom] NOUN Case=Nom|Number=Sing POSS 16 - - - - -
8 , , [Punct] PUNCT 7 - - - - -
9 Sagar Sagar [/N][Nom] PROPN Case=Nom|Number=Sing NE 10 - - - - -
10 Sarwar Sarwar [/N][Nom] PROPN Case=Nom|Number=Sing COORD 7 1:7 holo - - - - -
11 és és [/cnj] CCONJ 10 - - - - -
12 felesége feleség [/N][Poss.3Sg][Nom] NOUN Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=3 COORD 11 - - - - -
13 , , [Punct] PUNCT 12 - - - - -
14 Meherun Meherun [/N][Nom] PROPN Case=Nom|Number=Sing NE 15 - - - - -
15 Runi Runi [/N][Nom] PROPN Case=Nom|Number=Sing COORD 12 1:12 coref - - - - -
16 meggyilkolása meggyilkolás [/N][Poss.3Sg][Nom] NOUN Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=3 ATT 17 - - - - -
17 óta óta [/Post] ADP 3 - - - - -
18 , , [Punct] PUNCT 0 - - - - -
# sent_id = globv.1.conllup_2
1 A a [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 2 - - - - -
2 rendőrségnek rendőrség [/N][Dat] NOUN Case=Dat|Number=Sing DAT 7 - - - - -
3 a a [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 5 - - - - -
4 mai mai [/Adj][Nom] ADJ Case=Nom|Degree=Pos|Number=Sing ATT 5 - - - - -
5 napig nap [/N][Ter] NOUN Case=Ter|Number=Sing TIO 7 - - - - -
6 nem nem [/Adv] ADV Prontype=Neg NEG 7 - - - - -
7 sikerült sikerül [/V][Pst.NDef.3Sg] VERB Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act ROOT 0 - - - - -
8 olyan olyan [/Adj][Pro|Nom] PROPN Case=Nom|Number=Sing|Person=3|Prontype=Dem ATT 9 - - - - -
9 nyomokat nyom [/N][Pl][Acc] NOUN Case=Acc|Number=Plur OBJ 12 - - - - -
10 és és [/cnj] CCONJ 9 - - - - -
11 bizonyítékokat talált bizonyíték [/N][Pl][Acc] NOUN Case=Acc|Number=Plur COORD 10 - - - - -
12 találnia talál [/V][Inf.3Sg] VERB Number=Sing|Person=3|Tense=Pres|VerbForm=Inf|Voice=Act INF 7 - - - - - YES
13 , , [Punct] PUNCT 7 - - - - -
14 melyek amely [/N][Pro|Rel][Pl][Nom] PRON Case=Nom|Number=Plur|Person=3|Prontype=Rel SUBJ 16 2:9 rel - - - - -
15 segítségükre segítség [/N][Poss.3Pl][Subl] NOUN Case=Sub|Number=Sing|Number[psor]=Plur|Person[psor]=3 OBL 16 - - - - -
16 lehetnének van [/V][Mod|V][Cond.NDef.3Pl] VERB Definite=Ind|Mood=Cnd,Pot|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act ATT 7 - - - - - YES
17 az az [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 10 - - - - -
18 ügy ügy [/N][Nom] NOUN Case=Nom|Number=Sing POSS 19 1:16 coref - - - - -
19 megoldásában megoldás [/N][Poss.3Sg][Nom] NOUN Case=Ine|Number=Sing|Number[psor]=Sing|Person[psor]=3 OBL 16 - - - - -
20 , , [Punct] PUNCT 0 - - - - -
# sent_id = globv.1.conllup_3
1 A a [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 4 - - - - -
2 brutálisan brutális [/Adj][Manner/Adv] ADJ Case=Ess|Degree=Pos|Number=Sing MODE 3 - - - - -
3 meggyilkolt meggyilkolt [/Adj][Nom] ADJ Case=Nom|Degree=Pos|Number=Sing ATT 4 - - - - -
4 házaspárra házaspár [/N][Subl] NOUN Case=Sub|Number=Sing OBL 11 1:7 coref - - - - -
5 2012. 2012. [/Num|Digit][Ord/Adj][Nom][Punct] ADJ Case=Nom|Number=Sing|NumType=Ord ATT 6 - - - - -
6 február február [/N][Nom] NOUN Case=Nom|Number=Sing ATT 7 - - - - -
7 12-én 12 [/Num|Digit][OrdDate/N][Supe] NOUN Case=Sup|Number=Sing TLOCY 11 - - - - -
8 a a [/Det|Art.Def] DET Definite=Def|Prontype=Art DET 10 - - - - -
9 korai korai [/Adj][Nom] ADJ Case=Nom|Degree=Pos|Number=Sing ATT 10 - - - - -
10 órákban óra [/N][Pl][Ine] NOUN Case=Ine|Number=Plur TLOCY 11 - - - - -
11 találtak talál [/V][Pst.NDef.3Pl] VERB Definite=Ind|Mood=Ind|Number=Plur|Person=3|Tense=Past|VerbForm=Fin|Voice=Act ROOT 0 - - - - - YES
12 rá rá [/Prev] ADV 11 - - - - -

```

3.4. ábra. A KorKor.conllup fájlok felépítése.

feladatok előfeldolgozó lépéseként értelmezhető koreferenciafeloldás során csak a szövegben testesen szereplő elemek viszonyairól szeretne mondani valamit. Ekkor ahogy a bemenetként szolgáló szöveg nem tartalmaz zéró elemeket, úgy a kimenetben sem elvárt a zéró elemek megjelenése.

3.4. Nehézségek

A koreferencia annotálásakor számos nehézséggel találtam szembe magam, amelyek kezelésére a szakirodalom sem tudott megnyugtató választ kínálni.

Az 5. példa azt a problémát illusztrálja, amikor a referens állapota megváltozik (itt: meghal). Vajon a holttest koreferens az emberrel?

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

Az annotátorokkal egyetértésben voltunk abban a kérdésben, hogy valamiféle kapcsolatnak lennie kell a holttestek és az élő „előzményük” között a példamondatban, hiszen a szöveg olvasója tudja, hogy kiknek a holttestéről van szó. A hulla és a „tulajdonosa” között más szöveggörnyezetben azonban birtokos kapcsolat is szerepelhet, mely esetben viszont nem indokolt közöttük koreferencia-kapcsolatot feltételezni.

3.4. Nehézségek

Egy érdekes megfigyelést tettem a Wikipédia szócikkkel kapcsolatban, ami nehézségeket okozott a szócikk annotálása során is. A problémát a legjobban egy illusztratív példán keresztül tudom bemutatni. Ez a példa egy Wikipédia szócikk³², ami egy bizonyos fajtájú tyúkról szól. Először az állat általános jellemzőit ismerteti, hol él, mit eszik stb., majd áttér az állat különböző testrészeinek, tollazatának, tarajának leírására. A helyzet tovább bonyolódik, amikor külön leírást ad a tojó és a kakas jellemzőiről (egy-egy harmadik személyben). Ezek az esetek holonimaként vannak jelölve a korpuszban, ugyanakkor ez a megoldás vitatható. Ugyanaz a rész-egész viszony áll fenn egy állatfaj és a nő- vagy hímnemű egyedek között, mint az állatfaj és a rá jellemző testrészek között? És vajon ugyanaz a rész-egész viszony áll fenn egy állatfaj és egy konkrét példány között?

Érdekes kérdések merültek fel a színészekről, filmekről, sorozatokról szóló Wikipédia szócikkben, amelyekben a színész és az általa eljátszott szereplő is megjelenik. Az ehhez hasonló jelenségek kezelésére nem találtam útmutatót más koreferenciakorpuszokban, ezért fontosnak tartom itt megemlíteni.

Véleményem szerint a szerep és a színész nem koreferensek, hiszen más entitásra utalnak, a korpuszba beépített Wikipédia szócikkben (6. példa)³³ például Norbit a film univerzumának egyik résztvevőjére, Eddie Murphy pedig egy hollywoodi színészre. Ennek alapján a szerepet és az őt eljátszó színészt nem kötöttük össze a korpuszban. A Norbit esetében ráadásul tovább bonyolítja a helyzetet, hogy egy színész három szerepet is játszik a filmben. Mivel a különböző szerepek esetén is ugyanarra a színész-entitásra utalnak a színésznevek, így a különböző szerepeknél hivatkozott azonos színészneveket koreferensnek jelöltem.

(6) Norbit Albert Rice (Eddie Murphy) csecsemőként kerül az árvaházba, ahol Mr. Wong (Eddie Murphy) neveli fel őt a többi gyerekekkel. Ahogy cseperedik, megismerkedik Kate-tel (Thandie Newton), és a lány a legjobb barátja lesz, mindent együtt csinálnak, de hamarosan különválasztják őket, mivel Kate-et örökbe fogadják. Így a kis Norbitra rátalál egy testes pufi lány – Rasputia (Eddie Murphy), aki azonnal beleszeret, és nem hajlandó semmilyen körülmények között magára hagyni Norbitot. Felnőttként Norbit életében feltűnik a bombanővé érett gyerekkori szerelme, Kate. Norbit a két nő között vívódik: az egyik gyönyörű szép, a másik túlságosan el van hívva.

A 7. példa azt a nehézséget szemlélteti, amikor egy szó előzménye, amellyel koreferens, több tagból áll (*split antecedent*).

³²https://hu.wikipedia.org/wiki/Drezdai_ty%C3%BAk

³³<https://hu.wikipedia.org/wiki/Norbit>

3.5. Konverterek magyar morfológiai címkekészletek között

- (7) **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 a korpusz elkészültéig nem találtam megfelelő megoldást, amely beleillett volna az annotációs sémába.

A három kézi annotálást igénylő feladat – morfológiai egyértelműsítés, függőségi mondatelemzés, valamint anafora- és koreferenciaannotálás – közül épp a legutolsó volt a legnehezebb, ráadásul az anaforafeloldó kimenetének javítását az annotátorok több más alfeladattal együtt végezték el. Az annotálás során többször változott az annotálási útmutató és a címkekészlet, minden fájl tartalmazott vitás, megbeszélendő eseteket. A problémákkal kapcsolatos döntéseket az annotátorokkal közösen hoztuk meg. Ez a hosszú, sok konzultációt igénylő és aprólékos munkafolyamat jelzi, hogy a célzott jelenség nehezen annotálható, ez pedig az annotáció megbízhatóságát csökkenti. Sok koreferenciakorpusz készült már a világ nyelveire, mégsem találtam nyomát annak, hogy a nehézségeket, problémás eseteket összegyűjtötték volna. A korpusz építése során tett megfigyeléseink talán hasznosak lehetnek egy pontosabb annotációs séma elkészítéséhez.

3.5. Konverterek magyar morfológiai címkekészletek között

Amint a 3.3.6. fejezetben már említettem, a KorKor korpusz kétféle morfológiai címkekészletben is megjeleníti az egyértelműsített morfológiai elemzést. A KorKor esetében ennek az volt a motivációja, hogy átjárhatóságot biztosítsak az emTag által kibocsátott emMorph címkék és a dependenciaelemző által igényelt UD címkék között. Egy későbbi projekt, az NYTK-NerKor (Simon és Vadász, 2021) tervezésekor pedig célként tűztük ki, hogy nemzetközi sztenderdnek megfelelő címkekészletben (is) biztosítani szeretnénk az egyértelműsített morfológiai elemzést. Ezek az igények már korábban is felmerültek a napi feladatok során, ezért 2019-ben társszerzőmmel, Simon Eszterrel 2019-ben végzett kutatásunk (Vadász és Simon, 2019) során megvizsgáltuk a magyar morfológiai címkekészleteket, valamint konvertereket készítettünk hozzájuk. Emellett a KorKor építésekor az első nyelvi annotációs réteg, amelyet elkészítettünk, az egyértelműsített morfológiai elemzés volt, ráadásul a korpusz épí-

3.5. Konverterek magyar morfológiai címkekészletek között

téséhez el kellett készítenem az emmorph2ud konvertert. Ez a munka pedig igényelte a morfológiai címkekészletek alaposabb megismerését és megértését. Ebben az alfejezetben röviden ismertetem a morfológiai címkekészletekkel kapcsolatos munkát, a Simon Eszterrel közösen végzett panmorph projekt eredményeit. Az alfejezet (Vadász és Simon, 2019) alapján készült, a társszerző hozzájárulásával került a disszertációba.

3.5.1. Magyar morfológiai címkekészletek

Az egyes morfológiai címkekészletek között formai és elméleti különbségek is lehetnek. A formai különbségek viszonylag könnyen áthidalhatók, a címkekészletek összehasonlításakor az elméleti különbségek okozhatnak több fejtörést. Egyes annotációs sémák a szóalakban található összes morféma kódolására törekszenek, míg mások csupán az inflexiós morfémákat kódolják. Eltérések lehetnek a szófajkészletben, bizonyos alkategóriák használatában, valamint az egyes nyelvi jelenségek kezelésének finomságában is. Az ideális cél a veszteségmentes konverzió, amihez a működő megoldást a leginkább közelíteni kell.

A használatban lévő morfológiai annotációs sémákat és címkekészleteket vizsgálva azzal szembesültünk, hogy sok esetben kevésbé dokumentáltak, valamint hogy a közöttük működő konverterek jellemzően csak saját, belső használatra készültek. A címkekészletek eltérő dokumentáltságát egy nyilvános GitHub repozitórium³⁴ létrehozásával orvosoltuk, amely tartalmazza az egyes annotációk által alkalmazott címkék teljes listáját, valamint az általunk fejlesztett konvertereket. A tárhely könnyen bővíthető más, eddig nem vizsgált vagy újonnan létrejövő címkekészletek ismertetésével, illetve az ezekre fejlesztett konverterekkel.

A munka négy címkekészletet érintett. Az MSD (Morphosyntactic Description) (Erjavec, 2004) a magyarral együtt tíz nyelv részletes morfoszintaktikai reprezentációjára alkalmas. Különlegessége, hogy pozícióalapú kódolást valósít meg, vagyis a kód rögzített hosszúságú, és minden pozíciójához egy-egy morfoszintaktikai jegy van hozzárendelve, az egyes pozíciókat betöltő karakterek pedig a jegyekhez rendelt értékek. Az első pozíció mindig a szófaji kategóriáé, a többi pedig további morfoszintaktikai információkat kódol – például egy kijelentő módú, múlt idejű, egyes szám második személyű, tárgyas ragozású főige MSD-kódolásban így fest:

adtad ad Vmis2s---y

³⁴<https://github.com/dlt-rilmta/panmorph>

3.5. Konverterek magyar morfológiai címkekészletek között

Ez a szisztéma nem hierarchikus, vagyis nem tükrözi az egyes értékek közötti összefüggéseket, valamint a morfológiai jelöltséget sem, ám az alapos dokumentációból³⁵ kiderül, hogy melyek azok a kombinációk, amelyek előfordulhatnak az egyes címkekben, és melyek nem. Továbbá nem is sztring-alapú, ami azt jelenti, hogy sem a lemma, sem a morfológiai szegmentumok, sem az allomorfolk nem képezik részét a morfológiai elemzésnek. Nincsenek továbbá jelölve a derivációk sem, csak és kizárólag morfoszintaktikai kódok vannak.

A Szeged Korpusz és Treebank 1.0 (Csendes et al., 2004) és 2.0 változata (Csendes et al., 2005) MSD kódokat tartalmaz, valamint a magyarlanc 1.0 és 2.0 verziója is MSD kódokat adott ki. A magyarlanc 2.0-nak egy későbbi verziójában és a korpusz 2.5 változatában már a harmonizált MSD–KR kódkészlet található (Vincze et al., 2014), amely néhány tulajdonságában eltér az eredeti MSD kódolástól. A munkánk ezt a harmonizált változatot érintette, erre hivatkozom MSD-ként.

A Szeged Treebanknek létezik egy további verziója is, amely a 2009-es *Syntactic and Semantic Dependencies in Multiple Languages* című CoNLL shared task (Hajič et al., 2009) követelményeinek megfelelő felépítésű – ezt hívom CoNLL-nek. A CoNLL csak egy formátum, aminek a lényege, hogy a morfoszintaktikai információk linearizált jegy–érték párok formájában legyenek megfogalmazva, de az alkalmazott jegyek és lehetséges értékeik nem kötöttek. Ebben a változatban a CoNLL címkekészlet a Szeged Korpusz 2.0 MSD kódjaiból – tehát a még nem harmonizált MSD kódból – lett átalakítva.

A CoNLL kódolás az MSD kódot két részre osztja fel: az első pozícióban szereplő szófajkódot különválasztja, a további morfoszintaktikai információkat pedig a fent említett jegy–érték struktúrában jeleníti meg. Ebben a verzióban az egyes jegy–érték párok sorrendje kötött, az MSD pozícióit követi. Ha egy jegy nincs kitöltve értékkel, akkor *none* értéket kell, hogy kapjon. Az MSD-hez hasonlóan ez az annotációs séma sem tükrözi a morfológiai jelöltséget, továbbá erre is igaz, hogy sem a lemma, sem a morfológiai szegmentumok, sem az allomorfolk nem képezik részét a morfológiai elemzésnek. Nincsenek jelölve a derivációk sem, csak morfoszintaktikai kódokat tartalmaz. A fenti példa ebben a kódolásban így néz ki:

```
adtad ad V SubPOS=m|Mood=i|Tense=s|Per=2|Num=s|Def=y
```

A Szeged Dependency Treebanknek van egy olyan verziója is, amely a UD (Universal Dependencies and Morphology³⁶) nevű nemzetközileg elterjedt, univerzálisnak szánt annotációs séma szabályait követi (Vincze et al., 2016), valamint a magyarlanc 3.0 verziója is UD kódokat bocsát ki a

³⁵<http://nl.ijs.si/ME/Vault/V3/msd/msd.pdf>

³⁶<http://universaldependencies.org>

3.5. Konverterek magyar morfológiai címkekészletek között

morfológiai elemzés szintjén. A Szeged Dependency Treebank a UD 1. verziójának megfelelő címkeket tartalmazza. Azóta a UD 2. verziója is kijött már, de a magyar nyelvre és a Szeged Treebankre és így az azon alapuló eszközökre az újításokat automatikus módszerekkel vezették be. A UD kódolás sokban hasonlít a CoNLL-hez: ez is egy linearizált jegy-érték struktúrát valósít meg, de itt a jegyek ábécésorrendben szerepelnek, és az értékkel nem kitöltött jegyek nem jelennek meg. További tulajdonságaiban megegyezik a CoNLL fent ismertetett tulajdonságaival. A fenti példa ebben a kódolásban:

```
adtad ad VERB
Definite=Def|Mood=Ind|Number=Sing|
Person=2|Tense=Past|VerbForm=Fin|Voice=Act
```

A legújabb magyar morfológiai elemző az emMorph (Novák et al., 2016a), amely az emt sv morfológiai moduljaként is funkcionál. Ennek az elemzőnek az annotációs sémája jelentősen eltér az eddig ismertetettekétől, ugyanis sztringalapú, vagyis a lemma, a morfológiai szegmentumok és bizonyos esetekben az allomorfok is az elemzés részét képezik. További eltérést jelent, hogy nemcsak morfoszintaktikai információkat kódol, hanem olyan derivációkat is kezel, amelyeknek nem feltétlenül van köze az adott szó mondatbeli szerepéhez. Annyiban viszont hasonlít az MSD-hez, hogy nem hierarchikus, valamint nem tükrözi a morfológiai jelöltséget sem. Az emMorph többféle módon képes megjeleníteni a kimenetet aszerint, hogy tartalmazza-e a szóalakhhoz rendelt tövet és a szegmentumokat a szófajcímke és az elemzések mellett. Mi a tövet és a morfémákat nem tartalmazó morfológiai kódot konvertáljuk. A fenti példa ebben a rendszerben³⁷ ábrázolva:

```
adtad [/V] [Pst.Def.2Sg]
```

3.5.2. A konverterek

Legyen szó bármilyen formátumok közti konverzióról, többféle megközelítés létezik. Az egyik, ha a bemeneti címkekészletről a kimenetire egy közvetlen leképezést valósítunk meg. Egy másik lehetséges módszer, ha – a gépi fordítás egy fajtájánál használt *interlinguá*hoz hasonlóan – egy köztes metaformátumot találunk ki, amire le tudunk képezni minden bemeneti formátumot, és amiből elő

³⁷A címkek feloldása példákkal együtt az e-magyar honlapján (https://e-magyar.hu/hu/textmodules/emmorph_codelist) található.

3.5. Konverterek magyar morfológiai címkekészletek között

tudunk állítani minden kimeneti formátumot. Ez a magyar nyelv morfológiája esetében egy minden eddiginél részletesebb, a szokásos vitás kérdésekben (főnév vs. melléknév, inflexió vs. deriváció stb.) kötelezően döntést hozó, a morfológiai annotációk fent felsorolt tulajdonságait (hierarchikusság, sztringalapúság stb.) egyszerre birtokló újabb morfológiai annotációt eredményezne. Ezért az első megközelítés mellett döntöttünk, és közvetlen leképezést csináltunk három irányba, ahol a bemeneti oldalon mindig az emMorph kódja áll.

Az emMorph címkekészletről történő konvertálásnak több előnye is van. Egyrészt az emMorph formalizmusa összességében részletesebb, mint a célformalizmusok, ezért a konverzió viszonylag kis veszteséggel megoldható. Másrészt pedig a magyar nyelvre készült egyik legelterjedtebb elemzőláncba, az e-magyarba is az emMorph elemző van beépítve, így az e-magyarral elemzett szöveg tetszőlegesen átalakítható a kezelt címkekészletek valamelyikére a felhasználó céljainak megfelelően. Az emmorph2msd konverter kimenete a magyarlánc 2.0 által is előállított MSD kód; az emmorph2conll konverter kimenete az MSD kód átalakításával kialakított jegy-érték struktúrájú CoNLL kód; az emmorph2ud konverter kimenete pedig a magyarlánc 3.0 által is előállított UD kód.

A konverterek kidolgozásához megvizsgáltunk néhány elérhető konvertert, azok működéséből, felépítéséből levontuk a számunkra fontos tanulságokat. Az egyik ilyen konverter az e-magyarban működő DepTool.java³⁸, amely az emDep modul számára konvertálja az emMorph címkéket a fent ismertetett CoNLL formátumra, de egy belső, kevert címkekészletet használva. A magyarláncban is több konverter működik a címkekészletek között (pl. a harmonizált MSD és a UD között³⁹).

Az emmorph2ud konverter az e-magyar elemzőlánc emt_{sv} elnevezésű verziójában kiváltotta a DepTool.java konvertert. Az elemzőláncba illeszkedve az emMorph kimenetét konvertálja az emDep modul számára fogyasztható jegy-érték struktúrájú UD címkékre, valamint kimeneti formalizmusként lehetővé teszi, hogy a felhasználók az eddig elérhető emMorph kimenet mellett UD morfológiai címkéket is kaphassanak.

A konverterek elkészítésekor akkor volt a legkönnyebb dolgunk, amikor egy-az-egyhez megfeleltetés állt fenn a bemeneti és a kimeneti oldal között. Ugyanakkor sok esetben szükség volt a címkék megfeleltetésekor a eseteket és kivételeket megfogalmazni. Ennek oka a konverterek közötti elméleti különbségekben keresendő. Szemléltető példaként tekintsük a szófajok és az azokat reprezentáló címkék esetét. Az emMorph formalizmusában a szófajokat ábrázoló címkék megkülönböztetett for-

³⁸https://github.com/dlt-rilmata/hunlp-GATE/blob/master/Lang_Hungarian/src/hu/nytud/gate/util/DepTool.java

³⁹<https://github.com/zsibritajanos/magyarlanc/blob/master/magyarlanc/src/main/java/hu/uszeged/converter/univ/Msd2UnivMorph.java>

3.5. Konverterek magyar morfológiai címkekészletek között

mát kaptak a morfológiai jegyekhez képest ([/Adj]). Ugyanakkor a melléknevekhez és határozószóhoz járuló felsőfokot kifejező morféma is a szófajcímkékhez hasonló formátummal rendelkezik ([/Sup1]), így külön figyelmet kellett fordítanunk arra, hogy a felsőfokban álló melléknevek és határozószók szófaját kinyerjük. Ráadásul az emMorph a kimeneti címkekészletekkel ellentétben a derivációkat is megjeleníti a címkékben. A helyes konverzióhoz a legkülső képzett alak szófaját és az arra rakódó inflexiós jegyeket kellett kinyernünk az emMorph címkéből, és ezeket a jegyeket kellett a kimeneti címkekészletek megfelelő jegyeire konvertálnunk.

Elkerülhetetlen volt, hogy egyes esetekben a lemma vagy a token felszíni tulajdonságaira is támaszkodjunk a konverzió során. Bár az emMorph címkekészlete tűnik a legrészletesebbnek, néhány nyelvi jelenség esetében mégsem tartalmazza a helyes kimeneti címkéhez szükséges morfoszintaktikai vagy lexikai információt. Például a kötőszavak bizonyos tulajdonságait nem kódolja az emMorph, míg a UD, a CoNLL és az MSD is külön jegyet ad a mellérendelő és az alárendelő kötőszóknak. Emellett az MSD és a CoNLL az egyes és a páros kötőszókat is külön jeggyel választja ketté, valamint azt is jelöli, hogy mondatok vagy szavak között állnak az aktuális mondatban. Mivel ezeket az információkat nem kódolja az emMorph, ezért a biztosan egy csoportba tartozó kötőszók felsorolásával oldottuk meg a megfelelő kimeneti címke előállítását.

A névmások kezelésében is alapvető különbségek vannak az emMorph és a kimeneti címkekészletek között. Az MSD, a CoNLL és a UD szófajcímkéi között szerepel a névmási címke, kiegészítve a névmás típusát (személyes, mutató, kölcsönös, visszaható, általános stb.) reprezentáló információval. Az emMorph a névmások esetében a szófajcímkében azt tünteti fel, hogy milyen szófajú szó (főnév, melléknév, számnév, determináns vagy határozószó) helyettesítője. A névmástípusok közül csak a kérdő és a vonatkozó névmást jelöli a szófajcímkében. A névmások és azok típusai zárt szóosztályt alkotnak, így felsorolhatóak. Az emMorph által nem kezelt névmástípusok tagjainak felsorolásával igyekeztünk megoldani a helyes kimeneti címkék kinyerését a konverzió során.

Az igekötők kezelésében is találunk különbségeket. A UD a dokumentációk alapján csak a *meg* igekötőt jelöli külön szófajjal, a többi igekötőt eredeti szófaja alapján címkézi, így az emMorph által igekötőnek címkézett *meg* kapja csak az igekötőhöz tartozó szófajcímkét a UD-ra való konvertáláskor. A másik két kimeneti címkekészlet a többi igekötőt is igekötőként jelöli, így azokkal nem kellett külön foglalkoznunk.

A UD nem csak az igekötők kezelésében tér el a többi készlettől, hanem a tulajdonneveket is külön szófajcímkével látja el. Ezért amikor a lemmatizáló nagybetűs tövet tulajdonít a szóhoz, akkor a kimeneti szófajcímké az emMorph kódról konvertált főnévi címke helyett tulajdonnév lesz. Ekkor a helyes átalakítás a megfelelő tövesítésen múlik.

3.5. Konverterek magyar morfológiai címkekészletek között

Olyan jelenségek is akadnak, amelyek kimaradnak a konverzióból, vagyis hiába szerepelnek a kimeneti címkekészletben, a konverzió során nem tudnak előállni. Ez akkor fordul elő, ha a bemeneti oldalon nem szerepel egy jelenség, és a vizsgált szó felszíni tulajdonságaiból sem tudunk következtetni. Erre egy példa a birtokos eset címkéje. A magyarlánc a *-nAk* ragos névszók esetében mind a részesesetet, mind a birtokosesetet jelentő címkét tartalmazó címkesort kiadja, de az emMorph csak a datívuszi címkét ismeri, így a konverterünk is mindig csak *ilyet* fog kiadni. Egy hasonló példa a segédigék kezelése. A kimeneti címkekészletek megkülönböztetnek fő- és segédigéket, míg az emMorph nem.

A konvertereket Python3-ban implementáltuk. A kódok szabadon elérhetőek és felhasználhatóak GNU GPLv3 licenc alatt, míg a kódkészleteket ismertető dokumentációt és táblázatokat CC-BY-SA-4.0 licenc alatt publikáltuk⁴⁰.

3.5.3. Kiértékelés

Bár többféle mérést végeztünk, minden esetben csak a valós pozitív találatokat számoltuk össze, hiszen a feladat kiértékelésekor a fedésnek nincs értelme (minden címkét konvertálunk). Ezért csak pontosságot (*accuracy*) számoltunk oly módon, hogy a helyesen konvertált esetek számát elosztottuk az összes teszteset számával.

Háromféle tesztet végeztünk el. Az első – legmegengedőbb – teszt során azt ellenőriztük, hogy a konvertált címke előfordult-e valaha a magyarlánccal elemzett testanyagban (tehát sem a token, sem az emMorph címkét nem párosítottuk hozzá). Bár feltételezhetjük, hogy a testanyag ugyan nem tartalmazza az összes elképzelhető UD és MSD címkét, de a leggyakoribbakat biztosan, így ez a teszt annak a mérésére alkalmas, hogy valid címke jött-e létre a konverzió után. Vagyis ez csupán egy validitási kritériumot ellenőriz, önmagában nem elég mutatója a konverzió minőségének, elsősorban a fejlesztés során volt hasznos.

A második teszt volt a legszigorúbb, minden token esetében az ahhoz a tokenhez tartozó magyarlánc elemzésekkel vetettük össze a konvertált címkét. Emögött a mérőszám mögött az a feltételezés áll, hogy a kétféle elemző kimenetében szereplő címkék páronként megfeleltethetők egymásnak, mert ugyanaz a jelentésük. A valóságban azonban a két elemző sok jelenséget egészen eltérően kezel az annotációs sémák közötti elméleti különbségek miatt. Ráadásul az elemzők hibákat is vétének, ami szintén nehezíti az összehasonlítást. Ezzel a szigorú mérőszámmal tehát nem pusztán a konverziót

⁴⁰<https://github.com/dlt-rilmta/panmorph>

3.5. Konverterek magyar morfológiai címkekészletek között

értékeljük ki, hanem a kétféle elemző különbségeit is kidomborítjuk, mert olyan esetek is hibásnak számítanak, amelyek a kétféle elemző eltérő minőségéből vagy megközelítéséből adódnak. Ezeket a hibákat nem válogattuk szét, így az eredményeket ennek tudatában kell értékelni.

A harmadik tesztben – a fenti torzító hatást kiküszöbölendő – úgy számoltuk a pontosságot, hogy a tokenhez tartozó emMorph címkéről konvertált kimenetet nem a tokenhez tartozó gold standard – UD, MSD vagy CoNLL – címkével vetettük össze, hanem az összes olyan címkével, amely bármely, ugyanolyan emMorph elemzéssel rendelkező tokenhez tartozik. Például a [/N] [Pl] [Acc] emMorph címkéből konvertált kimeneti címkét azokkal a gold standard címkékkel vetjük össze, amelyek olyan tokenekhez tartoznak, amelyeknek szintén van [/N] [Pl] [Acc] elemzése. Ez egy megengedőbb kiértékelés, ugyanakkor feltehetőleg kiszűri a kétféle elemző különbségeinek torzító hatását. A konvertálók teljesítménye szempontjából ezt a mérőszámot tartjuk a legfontosabbnak.

3.5.4. Eredmények és diszkusszió

Az első teszt tehát azt vizsgálta, hogy valid címkék jönnek-e létre a konverzió során. A 3.5.4. táblázatban látható, hogy mindhárom konverter nagyon magas eredményeket ért el ezen a teszten, ám ez a magas szám alapvető elvárás, amely egy konverterrel szemben támasztható. Magyarázatra szorul azonban a tény, hogy egyik konverterrel sem sikerült elérni 100%-os eredményt. Mindhárom konverter esetében átnéztük a nem validnak ítélt címkék listáját, és ellenőriztük, hogy a rendelkezésünkre álló dokumentációk alapján hibásak-e. A leírások alapján megállapítottuk, hogy a nem validnak ítélt címkék valójában validak, csak egyszerűen hiányoztak a gold standard adatból.

	összes	TP	TN	ACCURACY
emmorph2ud	105 545	105 170	375	99,64%
emmorph2msd	105 545	104 539	1 006	99,05%
emmorph2conll	74 702	72 459	2 243	97,00%

3.8. táblázat. A konverterek eredményei az első teszten.

A 3.5.4. táblázatban ismertetett eredmények a második tesztre vonatkoznak, így az elvárásoknak megfelelően ezek a leggyengébbek. A kiinduló ötlet alapján ez lenne a megfelelő mérés a konverzió minőségére, ám szem előtt kell tartani a tesztek során tapasztalt torzító hatást, amelyet az egyes címkekészletek és az elemzők közötti alapvető elméleti különbségek okoznak. Gyakori például, hogy az egyik eszköz csak melléknévi, míg a másik csak főnévi címkét ad egy szónak. Még ha a többi jegyet

3.5. Konverterek magyar morfológiai címkekészletek között

sikeresen konvertálja is a konverter, és a konverzió valójában helyes kimeneti címkét eredményezett, amiatt, hogy az ennek megfelelő címke hiányzik a gold standard adatból, a konverzió is hibásnak számít. Ez a probléma akkor merül fel, ha az egyes emMorph elemzésekhez nem párosítható elemzés az összes magyarlánc kimenet közül, tehát amikor a magyarlánc fedése kisebb.

	összes	TP	TN	ACCURACY
emmorph2ud	105 545	87 506	18 039	82,91%
emmorph2msd	105 545	77 422	28 123	73,35%
emmorph2con11	74 702	52 176	22 526	69,85%

3.9. táblázat. A konverterek eredményei a második teszten.

Egy jellemző példa az anaforikus birtokos egyes és többes számú jelének előfordulása a fejlesztőanyagokban. A 3.5.4. táblázatból kiolvasható, hogy az emMorph szívesebben ad [AnP] és [AnP.P1] címkeket a névszóknak, mint a magyarlánc különböző verziói. Természetesen az emmorph2con11 kiértékelésekor ez a probléma fokozódik, mivel ott nem az összes lehetséges magyarlánc elemzés áll a rendelkezésünkre, hanem minden szóalakhhoz csak egyetlen, a korpuszban lévő egyértelműsített elemzés.

	anaforikus birtokosok
emMorph	8 959
MSD	1 136
UD	5 804

3.10. táblázat. Az emMorph és a magyarlánc két verziója által eredményezett egyes és többes számú anaforikus birtokosok darabszáma a fejlesztőanyagban.

A harmadik tesztet tekintjük a legalkalmasabb mutatónak a konverzió minőségére vonatkozóan. Az eredményeket a 3.5.4. táblázat ismerteti. Az emmorph2ud és az emmorph2msd konverterek esetében 97% fölötti eredményt értünk el, az emmorph2con11 azonban jóval gyengébben, bár 90% fölött teljesített a teszten.

Azt feltételezzük, hogy az emmorph2con11 gyenge eredményének az oka a kiértékelés módszerében keresendő. Míg az egyes tokenekhez az emMorph többféle elemzést is eredményezhetett, ad-

3.5. Konverterek magyar morfológiai címkekészletek között

	összes	TP	TN	ACCURACY
emmorph2ud	105 545	103 489	2 056	98,05%
emmorph2msd	105 545	102 693	2 852	97,30%
emmorph2conll	74 702	68 691	6 011	92,00%

3.11. táblázat. A konverterek eredményei a harmadik teszten.

dig az annotált korpuszban egy tokenhez természetesen jóval kevesebb elemzés tartozott. A 3.12. táblázat a Szeged Korpusz szólistájának token/címke arányát mutatja az emMorph-fal és a magyarlánc 3.0 verziójával megelemezve, valamint a CoNLL címkével annotált korpuszban. Minél magasabb ez a szám, annél több gold standard címkével tudjuk összevetni a konvertált címkét. Ez azt jelenti, hogy a harmadik teszt eredményét az emmorph2conll konverter kiértékelése esetében hasonló fenntartásokkal kell kezelni, mint a második teszt eredményeit.

	token	címke	címke/token
emMorph	152 056	293 956	1,93
UD	152 056	242 477	1,59
ConLL	152 056	159 033	1,05

3.12. táblázat. A Szeged Treebank címke/token arányai az egyes címkék szerint.

A különböző morfológiai címkekészletek közötti konverzió során felmerül az eltérő tövesítés problémája is. Az emMorph mint derivációt is kezelő morfológiai elemző nyilván más tövet fog megállapítani egy képzett szó esetében, mint azok az elemzők, amelyek csak az inflexiós jegyeket kódolják. A tesztanyagokon kimértük, hogy az esetek mekkora részében jelenik meg az eltérő tövesítés. Ha egy szóhoz akár a bemeneti, akár a kimeneti oldalon több elemzés társul a tesztanyagban, akkor nehézkes a tövesítés összevetése. Ezért közvetlenül nem tudjuk összehasonlítani az elemzőket, csak közvetett módon. Azoknál a szavaknál hasonlítottuk össze a töveket, ahol a fenti kiértékelés alapján a második tesztben a konverter hibátlanul konvertált a címkék között.

A 3.13. táblázatban látható eredmények azt mutatják, hogy mindhárom címkekészletpárt tekintve az esetek legnagyobb részében nem különböznek a tövek a helyesnek ítélt konverziók között. Természetesen ez az eredmény nem jelenti azt, hogy a morfológiai kódok közötti konverziókor nem kell foglalkozni az eltérő tövesítéssel, az itt ismertetett konvertálók azonban egyelőre csak a morfológiai címkék közötti átváltást vállalják.

3.6. A KorKor a CorefUD gyűjteményben

	TP	egyező tő	különböző tő	accuracy
emmorph2ud	87 506	80 237	7 269	91,69%
emmorph2msd	77 422	70 299	7 123	90,80%
emmorph2conll	52 176	48 021	4 155	92,04%

3.13. táblázat. Az egyező lemmák a helyes konverziók esetében.

3.6. A KorKor a CorefUD gyűjteményben

A Coreference in Universal Dependencies (CorefUD) projekt⁴¹ és a Universal Anaphora kezdeményezés célja, hogy összegyűjtse és harmonizálja a különböző nyelvekre készült koreferenciakorpuszokat mind az annotációs sémát, mind pedig a formátumot tekintve. A projekt jelenleg a CorefUD 1.1-es kiadásnál tart (Novák et al., 2022), 13 nyelvből tartalmaz 4 413 539 tokennyi anyagot 21 korpuszban. A nyelvi annotációk (morfológia, szintaxis) formátuma és sémái a Universal Dependencies (UD)⁴² keretrendszerébe illeszkednek, ezek mellett pedig a CorefUD projektben kifejlesztett annotációs sémákat alkalmazzák a koreferenciaannotációra.

A CorefUD korábbi kiadása tartalmazza a SzegedKoref anyagát, 2022 folyamán pedig megtörtént a KorKor felkészítése a CorefUD-hez való csatlakozáshoz. A KorKor szerencsére sok szempontból már eleve alkalmas volt a CorefUD-hez való csatlakozásra. Például a korpusz fájlformátumai, a CoNLL-U Plus és az `x.tsv` is átalakítható a CoNLL-U formátumra, valamint az UD szófajcímkék és morfológiai jegyek is megfelelőnek tűntek a harmonizáció számára. Ugyanakkor a KorKor-ban szereplő függőségi elemzés sémája és címkekészlete nem teszi lehetővé az UD-nek megfelelő konverziót, valamint a koreferenciaannotáció sémája is több tulajdonságában eltér az elvárttól, ezért nagyobb előkészítő munkát igényelt a konverzió. Noha a CorefUD tartalmaz olyan korpuszokat is, amelyek nem szabad lincenccel vannak közreadva, a projekt elsődleges célja a hozzáférhetőség biztosítása, ezért elsősorban a szabadon hozzáférhető korpuszok csatlakozását bátorítják, a KorKor pedig ilyen.

A KorKor integrációja a többi korpusz mintájára egy konverter szkript segítségével történt. A konverter bemenete a korpusz saját formátuma, ezért a korpuszban történő esetleges későbbi módosítások bekerülnek a CorefUD kiadásába is. A konverzió alapja a zérónévmásokat is tartalmazó változat volt, hiszen a UD lehetővé teszi a zéró elemek jelölését is.

⁴¹<https://ufal.mff.cuni.cz/corefud>

⁴²<https://universaldependencies.org/>

3.6. A KorKor a CorefUD gyűjteményben

Bár a KorKor morfológiai annotációja alkalmas lett volna arra, hogy egy az egyben beépüljön a CorefUD változatba, a projekt leírása alapján nem lehetett csak a morfológiai elemzési réteget importálni a függőségi elemzés nélkül. Mivel az utóbbi nem felel meg az UD sémájának, ezért azt a döntést hoztuk, hogy a korpusz automatikus morfológiai és szintaktikai elemzéssel ellátva kerül be a gyűjteménybe. Az elemzés elkészítéséhez az Udapi (Popel et al., 2017) elemzőt használtuk. Ha a későbbiekben a KorKor szintaktikai elemzési rétegét az UD előírásainak megfelelően frissítjük, akkor a CorefUD gyűjteménybe is bekerülhet a gold standard minőségű morfológiai elemzéssel együtt.

A koreferenciaannotáció szintén nagyobb módosításokat igényelt. A KorKor-ban eredetileg az anaforikus- vagy koreferenciakapcsolatban résztvevő elemek feje van jelölve, az összetevők határai nem. A CorefUD útmutatója alapján mind a fejnek, mind az összetevőhatároknak jelölve kell lenniük, ezért a függőségi elemzés alapján automatikusan történt az összetevők határainak megjelölése.

Az útmutató emellett elvárja, hogy az azonos referenciájú elemek azonos klaszterazonosítóval legyenek ellátva. Ezzel szemben eredetileg a KorKor-ban minden kapcsolatban részt vevő elem mellett az előző azonos referenciájú elem (vagy antecedens) indexe jelenik meg, tehát az annotáció koreferencialáncokat tartalmaz. Szerencsére a kétféle jelölés átjárható, ezért a KorKor annotációja automatikusan átalakítható volt a CorefUD számára alkalmas formátumra.

A harmadik fontosabb változtatás a címkéket érintette. A KorKor-ban használt típusok és címkékészlet helyett a CorefUD minden anaforikus kapcsolatot és minden olyan koreferenciakapcsolatot, ahol a kapcsolatban részt vevő elemek referenciája azonos, ugyanolyan címkével lát el (*Entity*) amellett, hogy közös klaszterbe sorolja őket. Emellett lehetőség van két további kapcsolattípus megjelölésére. A *Bridge* típus azon kapcsolatok címkézésére szolgál, ahol nem azonos a két elem referenciája, mégis valamilyen koreferenciakapcsolat (pl. rész-egész viszony) áll fenn közöttük. Szerencsére a KorKor ezeket külön címkével (*holo*) jelöli. A CorefUD sémájában használatos harmadik típus az ún. *split antecedens*ek kezelésére használatos (*SplitAnte*). Amint a 7. példában már rávilágítottam, a *split antecedens*ek kezelésére nem dolgoztam ki megfelelő annotációs elvet, ezért ezt a típust a KorKor CorefUD változata sem alkalmazza.

A CorefUD gyűjtemény csak a tanító és a fejlesztőadatot teszi közzé⁴³, a tesztadatot nem. Letöltés után használhatjuk a 13 nyelv 21 korpuszát egységes formátumban. A 3.5. ábrán a KorKor anyagából készült CorefUD korpuszfájl látható. Míg a KorKor eredetileg dokumentumonként egy korpuszfájlt tartalmaz, addig a CorefUD verzióban egy tanítófájl van. Az eredeti dokumentumhatá-

⁴³<https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-5053>

3.7. Összefoglalás

rokat a newdoc_id mező jelöli ki, emellett a detokenizált szöveg is megjelenik az egyes mondatok előtt.

```
# newdoc id = globv_1
# global.Entity = eid-etype-head-other
# sent_id = 109
# text = Három hónap telt el az újságíró házaspár, Sagar Sarwar és felesége, Meherun Runi meggyilkolása óta.
1 Három hónap NUM - Case=Nom|Number=Sing|NumType=Card 2 nummod 2:nummod -
2 hónap hónap NOUN - Case=Nom|Number=Sing 3 nsubj 3:nsubj -
3 telt tel VERB - Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act 0 root 0:root -
4 el el ADV - 3 compound:preverb 3:compound:preverb -
5 az az DET - Definite=Def|PronType=Art 7 det 7:det Entity=(e126--3|e127--3)
6 újságíró újságíró NOUN - Case=Nom|Number=Sing 7 nmod:att 7:nmod:att -
7 házaspár házaspár NOUN - Case=Nom|Number=Sing 3 nsubj 3:nsubj SpaceAfter=No -
8 , PUNCT - 9 punct 9:punct -
9 Sagar Sagar PROPN - Case=Nom|Number=Sing 7 appos 7:appos -
10 Sarwar Sarwar PROPN - Case=Nom|Number=Sing 9 flat:name 9:flat:name -
11 és és CONJ - 12 cc 12:cc -
12 felesége feleség NOUN - Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=3 7 conj 7:conj Entity=(e128--1|SpaceAfter=No)
13 , PUNCT - 14 punct 14:punct -
14 Meherun Meherun PROPN - Case=Nom|Number=Sing 16 nmod:att 16:nmod:att Entity=(e128--1)
15 Runi Runi PROPN - Case=Nom|Number=Sing 14 flat:name 14:flat:name Entity=(e128|e128)e127)
16 meggyilkolása meggyilkolás NOUN - Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=3 7 conj 7:conj Entity=e126)
17 óta óta ADP - 16 case 16:case SpaceAfter=No -
18 . PUNCT - 3 punct 3:punct -

# sent_id = 110
# text = A rendőrségnek a mai napig nem sikerült olyan nyomokat és bizonyítékokat találnia, melyek segítségükre lehetnének az ügy megoldásában.
1 A a DET - Definite=Def|PronType=Art 2 det 2:det Entity=(e129--2)
2 rendőrségnek rendőrség NOUN - Case=Dat|Number=Sing 7 iobj 7:iobj Entity=e129)
3 a a DET - Definite=Def|PronType=Art 5 det 5:det
4 mai mai ADJ - Case=Nom|Degree=Pos|Number=Sing 5 amod:att 5:amod:att -
5 napig nap NOUN - Case=Ter|Number=Sing 7 obl 7:obl -
6 nem nem ADV - PronType=Neg 7 advmod 7:advmod -
7 sikerült sikerül VERB - Definite=Ind|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act 0 root 0:root -
8 olyan olyan ADJ - Case=Nom|Degree=Pos|Number=Sing 9 amod:att 9:amod:att Entity=(e130--2)
9 nyomokat nyom NOUN - Case=Acc|Number=Plur 12 obj 12:obj -
10 és és CONJ - 11 cc 11:cc -
11 bizonyítékokat bizonyíték NOUN - Case=Acc|Number=Plur 9 conj 9:conj Entity=e130)
12 találnia talál VERB - Number=Sing|Person=3|VerbForm=Inf|Voice=Act 7 xcomp 7:xcomp SpaceAfter=No
12.1 DROP_SUBJ - 12:dep Entity=(e129--1)
13 , PUNCT - 16 punct 16:punct -
14 melyek mely PRON - Case=Nom|Number=Plur|Person=3|PronType=Rel 16 nsubj 16:nsubj Entity=(e130--1)
15 segítségükre segítség NOUN - Case=Sbl|Number=Plur 16 obl 16:obl -
15.1 DROP_POSS - 15:dep Entity=(e129--1)
16 lehetnének lehet VERB - Definite=Ind|Mood=Cnd|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act 12 acl 12:acl -
17 az az DET - Definite=Def|PronType=Art 18 det 18:det Entity=(e126--2)
18 ügy ügy NOUN - Case=Nom|Number=Sing 19 nmod:att 19:nmod:att Entity=e126)
19 megoldásában megoldás NOUN - Case=Ine|Number=Sing|Number[psor]=Sing|Person[psor]=3 16 obl 16:obl SpaceAfter=No
20 . PUNCT - 7 punct 7:punct -

# sent_id = 111
# text = A brutálisan meggyilkolt házaspárra 2012. február 12-én a korai órákban találták rá a Dhaka Nyugat Rajabazar városrészében lévő lakásukban.
```


3.5. ábra. A KorKor formátuma a CorefUD gyűjteményben.


Az Udapi vizualizációs eszközével megvizsgálhatjuk, hogy hogyan alakulnak a korpuszban a koreferenciakapcsolatok. A 3.6. ábrán egy részlet látható a korpuszból. Azonos színnel és klaszterazonosítóval az azonos referenciájú elemek vannak jelölve. A minden mondat mellett megjelenő fa ikonra kattintva a függőségi fákat lehet tanulmányozni.


3.7. Összefoglalás


A KorKor korpuszsal együtt publikált munkafolyamat és eszközök segíthetik további korpuszok elkészültét is. Az egyértelműsített tő és morfológiai címke alapvető nyelvi információ az annotált korpuszokban, ezért más projektek is célozzák ilyen annotációval ellátott korpuszok elkészültét. Az ELTE-DH gold standard korpuszának (K. Molnár és Dömötör, 2023) jelenleg is zajló munkálataihoz a KorKor projekt dokumentációit és eszközeit használják. Az egyértelműsített morfológiai címkék javításához a KorKor korpuszsal publikált munkafolyamat alapján fejlesztettek egy annotációs segédeszközt is.

3.7. Összefoglalás

 e260 Egy új német törvény értelmében, e260 mely októberben fog életbe lépni e260 DROP_SUBJ, e261 azoknak a közösségi oldalaknak büntetést kell majd fizetniük e261 DROP_SUBJ, e261 amelyek nem távolítják el a megjelenést követő 24 órán belül e262 a „kétségekívül törvénysértő” gyűlöletbeszédet.

 e260 A digitális jogok érvényesítését célzó törvény értelmében e261 a cégeknek 7 napjuk e261 DROP_POSS van arra, hogy döntsenek e261 DROP_SUBJ e262 a gyanús tartalmak sorsáról.

 A német büntető törvénykönyv már most is meghatározza e262 a gyűlöletbeszéd fogalmát, tehát e260 a törvény nem új intézkedést vagy fogalmat vezet be.

 Helyette arra kötelezi e260 DROP_SUBJ e261 a vállalatokat, hogy ellenőrizzék e261 DROP_SUBJ e262 a gyűlöletbeszédet, ellenkező esetben csillagászati összegű büntetést kell fizetniük e261 DROP_SUBJ.

3.6. ábra. A színes keretekben a koreferenciakapcsolatban résztvevő összetevők szerepelnek a klaszterek megjelölésével.

A KorKor anyagát és a munkafolyamatot magam is újra tudtam hasznosítani egy későbbi projektben, a 2021-ben zajló NYTK-NerKor korpusz⁴⁴ (Simon és Vadász, 2021) építésekor, aminek keretében egy egymillió szavas gold standard annotált korpusz készült. Az NYTK-NerKor projektet a 4. fejezet ismerteti részletesebben.

⁴⁴<https://github.com/nytud/NYTK-NerKor>

4. fejezet

NYTK-NerKor

Az NYTK-NerKor az egymillió tokenes méretével a jelenleg létező legnagyobb korpusz, amiben kézzel annotálták a névelemeket. A korpuszt társszerzővel, Simon Eszterrel készítettük. A projektben a hozzájárulásom elsősorban a morfológiai elemzéssel ellátott alkorpusz tervezése és elkészítése volt, majd a morfológiai elemzés kiterjesztése a teljes korpuszra.

Kiindulópontként a KorKor korpuszt használtam, hiszen formátuma, annotációs rétegei és egyéb tulajdonságai miatt alkalmas volt az újrafelhasználásra. A munkám magába foglalta a morfológiai címkék ellenőrzésével kapcsolatos alábbi feladatokat: az annotációs útmutató elkészítését, az annotátorok kiválasztását, a kapcsolattartást az annotátorokkal, az annotátorok által elkészített változatok összefésülését. Emellett az egyes formátumok között felmerült konverziós feladatokat is elvégeztem, valamint a morfológiai címkék konvertálásához használt emmorph2ud2 konvertert is elkészítettem.

A doktori munka kereteibe két szempontból is beleillik a munka. Egyrészt a korpusz tervezését megalapozta, építését pedig megkönnyítette a KorKor projektben felállított munkafolyamat, valamint a KorKor építéséhez készített eszközök. Másrészt pedig a névelemek lehetséges jelöltekként fontos résztvevői a szövegeket behálózó koreferenciakapcsolatoknak, a névelemannotált korpusz tehát erőforrása lehet egy, a koreferenciafeloldás előfeldolgozási lépését, a jelöltek kiválogatását végző eszköznek.

Ebben a fejezetben az NYTK-NerKor korpuszt, annak építési folyamatát és a kétféle annotációs réteg közül a morfológiai egyértelműsítés elkészítésének kérdéseit ismertetem. A fejezet (Simon és Vadász, 2021) és a projektdokumentáció alapján készült, a társszerző hozzájárulásával került a disszertációba.

4.1. Előzmények

4.1. Előzmények

A jelenleg elérhető magyar nyelvű gold standard tulajdonnév-annotált korpuszok közül a legismertebb a Szeged NER korpusz (Szarvas et al., 2006), amely gazdasági rövidhíreket tartalmaz, és összesen kb. 225 000 tokenből áll. A CoNLL2003 shared task (Tjong Kim Sang és De Meulder, 2003) annotációs sémáját és címkékészletét követi. A korpusz szövege a Szeged Treebankból (Csendes et al., 2005) alkorpusza, csak kutatási célokra lehet használni.

A másik a Criminal NE Korpusz¹, amely gazdasági bűncselekményekről szóló HVG-cikkekből áll, és kb. 560 000 tokent tartalmaz. Ez a korpusz a Magyar Nemzeti Szövegtár (Váradi, 2002) alkorpusza, ezért felhasználhatósága még inkább korlátozott. Ez is a CoNLL2003 címkékészletét követi.

A fent leírt gold standard korpuszok mellett létezik egy silver standard korpusz is. A hunNERwiki korpusz (Simon és Nemeskey, 2012) silver standard korpusz, hiszen automatikusan állították elő a magyar Wikipédiából. Az annotációs sémája megegyezik a Szeged NER korpuszéval, viszont több mint 19 millió tokenből áll. Szabadon felhasználható CC-BY-SA 3.0 licenc alatt.

A 4.1. táblázat összefoglalja a magyar névelemannotált korpuszokat. Az NYTK-NerKor nemcsak a legnagyobb, kézzel annotált ilyen korpusz, fontos tulajdonsága az is, hogy vegyes műfajokból tartalmaz szövegeket, valamint az, hogy szabadon felhasználható² CC-BY-SA 4.0 licenc alatt. Az NYTK-NerKor mérete összevethető az angol nyelvre a legismertebb hasonló korpusz, az OntoNotes 5.0³ méretével, amelynek az angol halmaza kb. 1,5 millió tokent tartalmaz.

	méret	műfaj	minőség
Szeged NER	225 000	gazdasági hírek	gold standard
Criminal NE Korpusz	560 000	gazdasági hírek	gold standard
hunNERwiki	19 000 000	Wikipedia	silver standard
NYTK-NerKor	1 000 000	vegyes	gold standard

4.1. táblázat. Az NYTK-NerKor a magyar névelemannotált korpuszok között.

A fentiek mellett fontos kiemelni, hogy a teljes korpusz kézzel ellenőrzött morfológiai egyértelműsítéssel is el lett látva. Mérete a Szeged Treebank (Csendes et al., 2005) méretével vethető össze,

¹<https://rgai.inf.u-szeged.hu/node/130>

²<https://github.com/nytud/NYTK-NerKor>

³<https://catalog.ldc.upenn.edu/LDC2013T19>

4.2. Műfajok

műfaj	fájl	mondat	token
fikció	122	24 690	203 014
jogi	39	7 272	191 984
hírek	82	9 767	213 157
webes	398	10 886	187 853
wikipédia	157	14 702	221 332
összesen	798	67 317	1 017 340

4.2. táblázat. Az NYTK-NerKor alkorpuszainak mérete.

amely kb. 1,2 millió tokent tartalmaz, szintén vegyes műfajokból válogatva, viszont ez a korpusz az MSD címkékészletet (Erjavec, 2004) címkékészletét használja. A Szeged Treebank engedélykérés után kutatási célokra használható fel.

4.2. Műfajok

Az NYTK-NerKor kiegyensúlyozott korpusz, öt műfajból tartalmaz egyenlő mennyiségű szöveget: fikció, jogi szövegek, hírek, webes szövegek és Wikipédia. Mivel a morfológiai jegyek nagy szerepet kapnak a hagyományos gépi tanuláson alapuló NER címkézők esetében, ezért a korpuszt elláttuk kézzel ellenőrzött morfológiai címkékkel is. A 4.2. táblázat a műfajok szerinti alkorpuszok méretét mutatja.

A cél egy referenciakorpusz elkészítése volt, ami alkalmas hagyományos és neurális gépi tanulási megoldások tanítására és kiértékelésére a névelemfelismerés területén, ezért a korpuszt tanító-, fejlesztő- és tesztalmazra bontva publikáltuk 80-10-10%-os arányban, így biztosítjuk a korpusz felhasználásával elért eredmények összehasonlíthatóságát. A 4.3. táblázat azt mutatja, hogy az egyes műfaji alkorpuszok hogyan oszlanak el erre a három halmazra.

4.3. Források

Fontos célunk volt, hogy a korpusz szabadon hozzáférhető legyen a felhasználók számára, ezért olyan forrásokból válogattunk szövegeket, amelyek biztosítják ezt. Egy másik fontos szempont volt, hogy lehetőleg minél frissebb, mai szövegeket gyűjtsünk. A KorKor korpuszt azért tudtuk felhasz-

4.3. Források

műfaj	tanító	fejlesztő	teszt
fikció	161 318	20 903	20 793
jogi	151 910	20 454	19 620
hírek	170 747	20 673	21 737
webes	150 725	18 401	18 727
wikipédia	176 515	22 667	22 150
összesen	811 215	103 098	103 027

4.3. táblázat. A tanító, fejlesztő és tesztadat tokenszámai az egyes alkorpuszokban.

nálni, mert ez a két fontos kritérium teljesül rá, ráadásul tartalmaz kézzel ellenőrzött morfológiai elemzést is. Mivel a KorKor két műfajból tartalmaz szövegeket (hírek és Wikipédia cikkek), ezért az NYTK-NerKor ezen alkorpuszaiba tudtuk beépíteni.

A fikciós szövegeket tartalmazó alkorpusz kétféle forrásból származik: szépirodalom és filmfeliratok. A szépirodalmi szövegek a Magyar Elektronikus Könyvtárból⁴ és a Project Gutenberg⁵ anyagából származnak, a filmfeliratok az OpenSubtitles⁶ gyűjtemény 2018-as részéből az Opus corpus⁷ felületén keresztül letöltve.

A jogi szövegek többféle EU-s forrásból származnak: egyrészt az EU Alkotmányból az Opus gyűjteményéből, az Európai Gazdasági és Szociális Bizottság 2017-es szövegeiből szintén az Opus gyűjteményéből, a JRC-Acquis (Steinberger et al., 2006) 2000 utáni részéből, valamint a DGT-Acquis (Steinberger et al., 2014) 2004 utáni részéből.

A hírek alkorpuszát is több forrásból gyűjtöttük: egyrészt az Európai Bizottság sajtóközlemény-adatbázisának 2016 utáni szövegeiből az Opus gyűjteményéből, másrészt a Global Voices hírportál magyar kiadásának 2010 és 2017 közötti szövegeiből az Opus gyűjteményéből (valamint egy része közvetlenül a KorKor korpuszból), harmadrészt pedig a NewsCrawl adatbázis (Barrault et al., 2019) 2019-es részéből.

A webes szövegek a Magyar Webkorpusz 2.0 (Nemeskey, 2020) 2017 és 2019 közötti anyagából származnak.

⁴<http://mek.oszk.hu/indexeng.phtml>

⁵<https://www.gutenberg.org/>

⁶<https://www.opensubtitles.org>

⁷<https://opus.nlpl.eu/index.php>

4.4. Az annotáció formátuma

A Wikipédia alkorpusz a magyar Wikipédiából tartalmaz cikkeket: egyrészt a KorKor korpuszból, másrészt pedig a hunNERwiki korpuszból (Simon és Nemeskey, 2012), ami tartalmaz silver standard névelem címkéket. Utóbbiból azokat a mondatokat válogattuk ki, amelyek legalább egy névelemet tartalmaznak.

4.4. Az annotáció formátuma

A korpusz CoNLL-U Plus formátumban van. Minden fájl hat oszlopot tartalmaz, az oszlopokat a fájl első sorában szereplő sor határozza meg:

```
# global.columns = FORM LEMMA UPOS XPOS FEATS CONLL:NER,
```

ahol:

FORM a token;

LEMMA a token lemmája;

UPOS az UD szófajcímkéje;

XPOS a teljes morfológiai címke (POS + morfoszintaktikai jegyek) az emMorph címkékészletével;

FEATS az UD morfoszintaktikai jegyek;

CONLL:NER névelemannotáció.

4.5. A morfológiai annotálás első üteme

Eredetileg a korpusz egyötöde kapott morfológiai annotációt. Kétféle morfológiai annotációt állítottunk elő: egyrészt megtartottuk az emMorph által kiadott elemzést, másrészt ezt átkonvertáltuk a UD jelenleg elérhető aktuális morfológiai formalizmusára (v2). Az emMorph által kibocsátott lemma került a LEMMA oszlopba, a morfológiai elemzést tartalmazó címke pedig az XPOS oszlopba. Az emMorph címkéből történő konverzió eredményeként kapott tő a UPOS oszlopba, a morfoszintaktikai jegyek pedig a FEATS oszlopba kerültek.

4.5. A morfológiai annotálás első üteme

Az `emt sv` elemzőnek létezik egy `emmorph2ud` modulja⁸, amely azonban a UD 1-es verziójának megfelelő morfológiai annotációt ad ki⁹. Az `emmorph2ud`-t alapul véve készítettem egy konvertert, amely a UD 2-es verzióját adja kimenetként. Ez `emmorph2ud2` néven új modulként bekerült az `emt sv` elemzőbe, valamint egy külön GitHub repozitóriumban is elérhető¹⁰.

4.5.1. A szövegek kiválasztása

A morfológiai annotációval is ellátott szövegek két forrásból származtak. Mivel a KorKor megfelelő annotációs rétegei ugyanezen annotációs elvek mentén lettek elkészítve, így a KorKor gold standard (két annotátor által annotált, majd egy harmadik által ellenőrzött) alkorpusza további feldolgozás nélkül beépíthető volt az NYTK-NerKorba, ezeket a szövegeket egyből továbbírányítottuk a NE annotálási fázisba. A KorKor azon részét, amelyet még csak egy annotátor és egy ellenőrző annotátor nézett végig, jelen projekt keretében még egy annotátorral annotáltattuk, majd ellenőriztük.

A szövegek másik forrása a Magyar Webkorpusz 2.0 (Nemeskey, 2020) volt, ami automatikusan gyűjtött egyes webes szövegeket tartalmaz, és eleve tartalmaz morfológiai elemzést.

4.5.2. Előfeldolgozás

A KorKorból származó szövegek már átestek a sztenderd előfeldolgozási lépéseken, vagyis mondatokra és tokenekre lettek bontva, valamint kaptak egyértelmű morfológiai elemzést. A Webcorpus 2.0 anyagai eleve fel voltak dolgozva az `emt sv` elemzővel. Így se a KorKorból, se a Webcorpus 2.0-ból származó szövegek esetében nem volt szükség külön előfeldolgozásra. Az `emt sv` kimenetét, az `xt sv` fájlokat úgy módosítottuk, hogy megfelelő legyen egy az annotációt segítő, ergonomikus annotációs környezet kialakításához.

⁸Az `emmorph2ud` az `emDep` számára megfelelő bemenetet készít, mivel a függőségi elemző tanítóadata is a UD 1-es verziójának címkékészletével készült.

⁹A UD v1 és v2 közötti különbségeket a <https://universaldependencies.org/v2/summary.html> oldal foglalja össze.

¹⁰<https://github.com/vadno/emmorph2ud2>

4.5. A morfológiai annotálás első üteme

4.5.3. Az annotációs folyamat és felület

Az annotációs folyamat, a feladat meghatározása és az annotációs felület megegyezett a 3.3.4. alfejezetben leírtakkal, tehát az annotátorok feladata az volt, hogy az `emt_sv` `emTag` moduljának a kimenetét ellenőrizzék és javítsák. A manuális feldolgozás után ismét `xtsv` formátumú adatot kaptunk, amelynek mezői megegyeztek az annotálás előtti állapottal, viszont az egyes tokenekhez már a javított `tő` és `emMorph` címke tartozott.

A morfológiai annotáción összesen 6 annotátor dolgozott, mindannyian nyelvész végzettségűek. Közülük ketten már végezték ugyanezt a feladatot a KorKor építése során is. Az annotátorokkal kizárólag virtuálisan, de folyamatosan tartottuk a kapcsolatot – virtuális megbeszélések és egy erre a célra létrehozott levelezőlista segítségével. A menet közben felmerülő problémás kérdéseket megvittattuk, a tanulságokkal bővítettük az annotációs útmutatót. A fájlkiosztás során igyekeztünk arra is figyelni, hogy az annotátorpárok egyenletesen oszoljanak el, vagyis ne mindig ugyanazok a párok annotálják ugyanazt a szöveget, így egyenletesebb lett a minőség is. Az annotátorok átlagos haladási tempója 1 074 token/óra volt.

4.5.4. Annotátorok közötti egyetértés

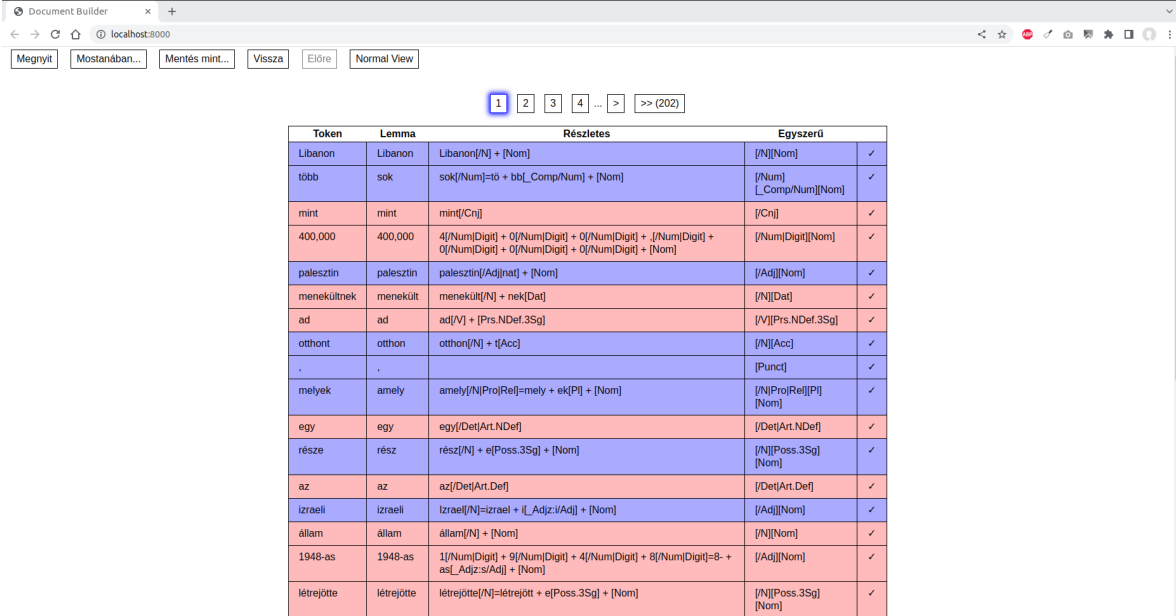
Az annotátorok közötti egyetértés az első 200 000 tokenen kiszámolva, Cohen κ mérőszámával kifejezve a lemmatizálást tekintve 97,6% lett, a teljes (UD) morfológiai címkét (szófaj + morfoszintaktikai jegyek) tekintve 94,52% az összes dokumentumra és az annotátorokra átlagolva. Az egyetértési eredményeket magyar korpuszokkal érdemes összevetni, azonban sajnos a Szeged Treebank megjelenésekor nem közöltek egyetértési értékeket. A KorKor a 3.3.4. alfejezetben már ismertetett eredményei valamivel magasabbak lettek (97% a teljes (UD) morfológiai címkére Krippendorff alfában kifejezve). Az összevetést nehezíti, hogy az egyetértés a két korpusz esetében eltérő mérőszámmal van kifejezve, ugyanakkor mindkét korpusz eredményei nagyon magas egyetértési arányt mutatnak. Ráadásul az NYTK-NerKor esetében ötféle műfajból származó szöveget kellett annotálni, amelyek körül az annotátorok beszámolóí alapján a filmfeliratokban és a webes szövegekben volt a legnehezebb. Ezekben a szövegekben sok olyan szó volt, amit a morfológiai elemző nem tudott helyesen tövesíteni, és sok esetben előfordult, hogy a két annotátor közül csak az egyik szűrte ki a hibát.

4.6. A morfológiai elemzés második üteme

4.6. A morfológiai elemzés második üteme

2022-ben NYTK-NerKor korpusz anyagának a maradék négyötöde is kapott egyértelműsített morfológiai címkéket. Az így előálló korpusz alkalmas arra, hogy morfológiai elemzőt tanítsanak rajta. A korpusz egyedülálló két szempontból is: egyrészt a mérete miatt, mert az egymillió tokenes méretével kiemelkedik a kézzel annotált korpuszok közül, másrészt pedig a címkekészletek miatt, amelyeket alkalmaztunk. Az emMorph a legfinomabb és legrészletesebb címkekészlet magyarrá, a UD pedig nemzetközileg elterjedt készlet.

Ennek a nyolcszázezer tokennek az annotálását szövegenként szintén két annotátor végezte. Az annotáláshoz ezúttal (K. Molnár és Dömötör, 2023) épülő korpuszához készített annotációs eszközt, a DocBuilder programot használtuk. A 4.1. ábrán látható a felület, amelyen az annotátorok dolgoztak.



Token	Lemma	Részletes	Egyszerű	
Libanon	Libanon	Libanon/[N] + [Nom]	[N][Nom]	✓
több	sok	sok/[Num]=tő + bb[_Comp/Num] + [Nom]	[/Num] [_Comp/Num][Nom]	✓
mint	mint	mint/[Cnj]	[/Cnj]	✓
400.000	400.000	4/[Num][Digit] + 0/[Num][Digit] + 0/[Num][Digit] + [Num][Digit] + 0/[Num][Digit] + 0/[Num][Digit] + 0/[Num][Digit] + [Nom]	[/Num][Digit][Nom]	✓
palesztin	palesztin	palesztin/[Adj][rat] + [Nom]	[/Adj][Nom]	✓
menekültek	menekült	menekült/[N] + nek[Dat]	[N][Dat]	✓
ad	ad	ad/[V] + [Prs.NDef.3Sg]	[/V][Prs.NDef.3Sg]	✓
otthon	otthon	otthon/[N] + [Acc]	[N][Acc]	✓
,	,	,	[Punct]	✓
melyek	amely	amely/[N][Pro][Rel]=mely + ek[Pl] + [Nom]	[/N][Pro][Rel][Pl] [Nom]	✓
egy	egy	egy/[Det][Art.NDef]	[/Det][Art.NDef]	✓
része	rész	rész/[N] + e[Poss.3Sg] + [Nom]	[N][Poss.3Sg] [Nom]	✓
az	az	az/[Det][Art.Def]	[/Det][Art.Def]	✓
izraeli	izraeli	izrael/[N]=izrael + [_Adj]:i[Adj] + [Nom]	[/Adj][Nom]	✓
állam	állam	állam/[N] + [Nom]	[N][Nom]	✓
1948-as	1948-as	1/[Num][Digit] + 9/[Num][Digit] + 4/[Num][Digit] + 8/[Num][Digit]=8- + as[_Adj]:s[Adj] + [Nom]	[/Adj][Nom]	✓
létrejött	létrejött	létrejött/[N]=létrejött + e[Poss.3Sg] + [Nom]	[N][Poss.3Sg] [Nom]	✓

4.1. ábra. A morfológiai egyértelműsítés ellenőrzése a DocBuilder annotálóprogramban.

Minden token sorában a pipa ikonra kattintva lehet elfogadni az emTag által kibocsátott elemzést és tövet, amelyek a Lemma' és az 'Egyszerű' elnevezésű oszlopokban láthatók. A pirossal kiemelt sorok olyan szóalakokat jelölnek, ahol többféle alternatív elemzési lehetőséget is felkínált az emMorph, így ezekre különös figyelmet kell fordítani. A címkére kattintva megtekinthetjük az emMorph által felkínált többi elemzési lehetősége és tövet, így ki tudjuk választani az aktuálisan helyeset. Ha egyik sem felel meg a kontextusnak, akkor kézzel is megadhatjuk a tövet vagy a címkét.

4.6. A morfológiai elemzés második üteme

A felület lehetőséget biztosít a tokenizálási hibák javítására is: össze lehet vonni két szomszédos tokent, szét lehet szedni egy tokent ketté, valamint token beillesztésére is van mód. Ezen felül a mondatszegmentálást is módosíthatjuk.

A beillesztett tokenek vagy javított szóalakok esetében a program azonnal leelemzi az új szóalakat, hiszen össze van kötve az emMorph elemzővel, így az eredeti szövegben még nem szereplő szóalakok esetében sem kell feltétlenül teljesen a nulláról felépíteni a címkét az annotátoroknak.

A két annotátor által címkézett szövegek ellenőrzését és javítását egy harmadik nyelvész és jómagam végeztük. A két annotátor által kezelt fájlokat összefésülésük egymás mellett jelenítettük meg. Erre egy olyan programot¹¹ használtunk, ami kiemeli a két fájl közti különbségeket. A két annotátor által eltérően címkézett vagy tövesített tokenek esetében mindenképpen döntést kellett hozni, azonban az átnézéskor azokra az esetekre is figyelni kellett, amikor ugyan egyformán dolgozott a két annotátor, de mindketten hibásan. Az utóbbi hibatípust nehezebb kiszűrni, hiszen a vizuális összevető nem tudja kiemelni. Emiatt az ellenőrzéskor mindenképpen végig kellett nézni minden egyes tokent és megtalálni a lehető legtöbb egyezőt, de hibás címkét és tövet.

Az ellenőrzés során kiderült, hogy melyek azok a jelenségek és elemzések, amelyek a leggyakrabban egyformán helytelenek mindkét annotátornál. Ezekben az esetekben jellemzően arról van szó, hogy az annotátorok közül egyik sem vette észre az előelemző által vétett hibát. Az alábbi jelenségek okozták a legtöbb ehhez hasonló hibát:

Gyakran előfordult a [/V][Prs.NDef.2Pl] címke azokon a helyeken, ahol [/V][Pst.NDef.3Pl] lenne a helyes. A szövegekben a legtöbb ige harmadik személyben van, ritkábban fordulnak elő második személyű alakok (elsősorban a filmfeliratokban, a szépirodalmi és a webes szövegekben). Ugyanakkor érzékelhető, hogy az emTag ezeknek a nyelvtanilag homonim esetekben (pl. *végezték*) hajlamos a második személyű ragozást választani. Mivel a többi számban és személyben nem áll fenn ez a jelenség, ezért az annotátorok gyakran nem vették észre ezeket a hibákat.

A kijelentő mód, jelen idejű, definit, többes szám harmadik személyű igealakok homonimák lehetnek a kötőmódú, definit, többes szám harmadik személyű alakokkal, viszont az annotátorok gyakran nem vették észre, ha nem a megfelelő címke járult ezekhez az igékhez.

¹¹Alkalmassak a feladatra pl. a Meld (<https://github.com/GNOME/meld>) és a WinMerge (<https://winmerge.org/>) vizuális összevető programok.

4.6. A morfológiai elemzés második üteme

Az emTag tanítóadatában minden melléknévi igenév melléknévnek, a határozói igenevek pedig határozószónak volt címkézve, ezért az emTag kimenetében nem jelennek meg a melléknévi igenév-képzők és a határozói igenév-képzők. Természetesen nemcsak a melléknévi és a határozói igenév-képzők jártak így, az emTag kimeneti címkekészlete jóval kevesebb képzős címkerészletet tartalmaz, mint az emMorph készlete. Az NYTK-NerKor tervezésekor úgy döntöttem, hogy a melléknévi és a határozói igenevek jelenjenek meg a morfológiai címkékben. Ennek elsősorban az az oka, hogy a UD morfoszintaktikai jegyei között szintén meg kell jelennie ezeknek a jegyeknek. Ha a felhasználó az emTag-nak megfelelő címkekészlettel szeretné használni a korpuszt, akkor egy egyszerű előfeldolgozó lépésben át tudja alakítani ezeket a címkéket melléknévi és határozószói címkékre, ugyanakkor fordítva ez nem lehetett volna lehetséges.

A *volna* kétféle igei címkét kell, hogy kapjon annak megfelelően, hogy segédige szerepében áll-e, vagy sem. Mind az emMorph, mind a UD címkekészlete tartalmaz segédige szófajt vagy segédige szerepre utaló címkerészletet, az annotátorok figyelmét azonban gyakran elkerülte.

A tövesítés javítása nehézséget okozott a létigék és kopulák esetében. A *van-*, *vol-*, *vagy-* kezdetű szóalakok töve *van*, a *le-*, *lesz-* kezdetű szóalakok töve *lesz* az útmutató szerint, a *lehet* töve *lehet*, ha modális segédige.

A *ez*, *az* tövű névmási determinánsok és főnevek elkülönítése is sok gondot okozott, különösen azért, mert úgy tűnik, hogy az emTag kimenete (és a tanítóadat annotációja) sem volt következetes. Az útmutató alapján ha a szó önmagában egy főnévi csoportot alkot, akkor a főnévi címkét ([/N] [Pro]), ha pedig determinánsi szerepű egy nagyobb főnévi csoportban, akkor a determinánsi címkét ([/Det] [Pro]) kellett választani. Sajnos az annotátorok gyakran átsiklottak a különbség fölött.

Az esetragos vonzattal rendelkező névutók és határozószók címkéje gyakran okozott nehézséget, különösen azok a névutók, amelyek birtokos szerkezetet alkotnak a vonzatukkal (*számára*).

Az emTag kimenetében gyakrabban jelenik meg az anaforikus birtokos címkéje, mint a szövegbeli valós előfordulása, és az annotátorok is gyakran átsiklottak felette. Ezek az alakok gyakran homonimák, pl. a *könnyebbé* szó elemzése tartalmazhat anaforikus birtokost vagy transzlatívusz esetragot.

A törzszámnevek töve rendszeresen helytelen az emTag kimenetében (pl. az *egykedtet* töve *egykét*), viszont az annotátorok gyakran nem vették észre.

4.7. Összefoglalás

Egy gyakori szó gyakran több értelmes elemzéssel is rendelkezhet, amelyek között nehéz dönteni, illetve könnyű átsiklani a lehetőségek fölött. Az *akkor* szóalak négyféle elemzést is kaphat, és a négyféle elemzés közül gyakran nem is olyan egyszerű választani:

- feltételes mondatban, ha nem időre vonatkozik [/Cn j] (*Ha felkészülsz, akkor nem lesz gond.*)
- ha főnévre utal [/N | Pro] [Temp] (*Szeretem a karácsonyt, mert akkor együtt van a család.*)
- határozószóként [/Adv]
- vonatkozó névmásként [/Adv | Pro]

Mivel az NYTK-NerKor fő annotációs rétege a névelemcímkézés, ezért külön figyelmet kell fordítani a tulajdonnevek elemzésére. A korpusz tervezésekor azt a döntést hoztuk, hogy még a szerkezettel bíró tulajdonnevek esetében sem társítunk morfológiai elemzést a nevekhez. A tulajdonnevek minden nem utolsó szava esetrag nélküli főnévi címkét ([/N]) kaptak, az utolsó (vagy egyetlen) szó pedig esetragos címkét. A tulajdonneveket alkotó szavak tövei nagybetűsek, ha a szóalak is nagybetűs.

Gyakran talákoztunk tokenizálási hibákkal is, de külön említést érdemelnek a kijelentő mondat végére eső rövidítések. Ebben az esetben az útmutató szerint a szóalakon nem jelenik meg a rövidítés pontja, a pont külön tokenként, mondatzáró pontként szerepel. A rövidítés szótövén azonban megjelenítjük a rövidítést jelentő pontot.

A fenti felsorolásból kiderül, hogy a homonimák okozták a legtöbb olyan hibát, amelyeket gyakran egyik annotátor sem vett észre. A két annotátor által kezelt fájlok összefésülésekor az ehhez hasonló, gyakran előforduló esetekre külön rákeresünk a fájlokban, hogy biztosan egységesen és helyesen legyenek kezelve. A vizuális összevető által kiemelt különbségek esetében döntést hozunk, de emellett a fájl összes tokenjét is leellenőrizzük a ritkább, nem rendszeresen előforduló hibák kiszűréseért.

4.7. Összefoglalás

Azzal, hogy az NYTK-NerKor egymillió tokenje kézzel javított morfológiai egyértelműsítést kapott, a korpusz alkalmassá vált arra, hogy tanítóanyagként használjuk fel egy újabb morfológiai egyértelműsítő elkészítéséhez. Így az NYTK-NerKor már nem csak a névelemfelismerés, hanem a morfológiai egyértelműsítés területén is fontos erőforrás lett.

4.7. Összefoglalás

A kézi javítás tanulságai azt a célt is szolgálhatják, hogy a rendszeresen előforduló hibákat könnyebben javíthassuk az emTag kimenetében, illetve hogy az emTag tanítóanyagaként használt korpuszban megvizsgáljuk és javítsuk a rendszeres hibákat okozó eseteket.

5. fejezet

Adatbázisok a névmási anaforafeloldáshoz

Ebben a fejezetben olyan adatbázisokat mutatok be, amelyek a többértelmű névmási anaforák feloldásának problémája köré szerveződnek. Ezekben az adatbázisokban és sémagyűjteményekben közös, hogy mind olyan mondatokat, mondatpárokat vagy rövid szövegeket tartalmaznak, amelyekben jelen van legalább egy többértelmű névmási anafora, amelynek feloldásához nem elegendő grammatikai és lexikális információkra támaszkodni, szükség van hozzá világismeretre és következtetési képességre. Éppen emiatt fontosak és aktuálisak ezek az erőforrások: a többértelmű névmási anaforák feloldása indikátora lehet annak, hogy egy számítógépes nyelvmodell mennyire képes megérteni a szövegben kódolt tartalmat.

A Winograd-sémák eredeti ötletének ismertetése után bemutatom, hogy hogyan lehet őket névmási anaforafeloldó eszközök tesztelésére használni, majd áttérek azokra a korpuszokra, amelyek a Winograd-sémákhoz köthetők, de magyar nyelvre készítettük őket. A fejezetben bemutatott magyar sémákat ismertető publikációt ([Vadász és Ligeti-Nagy, 2022](#)) társszerzővel, Ligeti-Nagy Noémivel készítettem, a fejezet az ő hozzájárulásával került a disszertációba. A projekthez az alábbi feladatok elvégzésével járultam hozzá: a Winograd-sémákhoz hasonló gyűjtemények felkutatása; a gépi fordítás ellenőrzése és javítása a sémák nagy részén; az összes létrejött magyar séma ellenőrzése; a PWS elkészítése (a hét fordítás összefésülése sémánként); a HuWS, a PWS és a HAPP gyűjtemények végleges formátumra történő átalakítása.

5.1. A Winograd-sémák

5.1. A Winograd-sémák

Hirst (1981) az automatikus anaforafeloldásról szóló alapmunkájában mondatpárokkal illusztrálja az anafora jelenségét. A Winograd-sémák is mondatpárok, ahol a két mondat tartalma egy összetevőt kivéve teljesen megegyezik, de a két mondatban a lexikálisan azonos célnév-mások különböző antecedenssel rendelkeznek. A Winograd-sémák azt illusztrálják, hogy a névmások antecedensének meghatározásához gyakran nem elég a grammatikai információ (pl. kötéselemek), a világismeret és a következtetés (*commonsense reasoning*), éppen ezért az anaforafeloldás magas szintű, komplex nyelvfeldolgozási feladat – számítógép és ember számára egyaránt.

Amint a 8. példában¹ látható, ahhoz, hogy helyesen jelöljük ki a névmás antecedensét – legyen szó a humán nyelvfeldolgozásról vagy a gépi anaforafeloldásról – a mondatok grammatikai feldolgozása mellett tudnunk kell például azt is, hogy általában a városi tanácsstagok tartanak az erőszakotól és a tüntetők azok, akik az erőszakot támogatják (legalábbis a városi tanácsstagok szerint).

(8) A városi tanácsstagok nem adtak engedélyt a tüntetőknek, mert [kerülték/támogatták] az erőszakot.

Kik [kerülték/támogatták] az erőszakot?

- a. a városi tanácsstagok
- b. a tüntetők

A sikeres antecedenskeresés többféle tudást igényelhet, legyen szó társas, kulturális tudásról, lexikális ismeretekről, a világban szerzett összes tapasztalatunkról – ezekre mind világismeretként hivatkozunk –, de ide tartozik a hallgató és a beszélő által birtokolt közös tudás is.

5.1.1. A Winograd-séma kihívás mint az új Turing-teszt

A Turing (1950) által bevezetett teszt a gépi intelligenciát vizsgálja. A teszt egy imitációs játékként is elképzelhető, ahol egy kérdező kérdéseket tesz fel két résztvevőnek, akik közül az egyik egy humán válaszadó, a másik pedig egy számítógép. A kérdező tudja, hogy az egyik válaszoló fél egy számítógép, de nem tudja, hogy melyik. A számítógép célja, hogy meggyőzze a kérdezőt arról, hogy ő a humán résztvevő. Turing azt jósolta, hogy a XX. század végére a kérdezőknek csak 70% esélye lesz rá, hogy helyesen ítélik meg a válaszolók kilétét ötpercnyi faggatózás után (Turing, 1950, 442. oldal).

¹A példamondat Terry Winograd példájának magyar fordítása.

5.1. A Winograd-sémák

A „Turing-teszt” általános kifejezésként is használatos bármilyen feladatra vagy tesztre, ami a számítógép intelligenciáját méri. Az imitációs játék vitákat generált azzal kapcsolatban, hogy hogyan definiáljuk a számítógépek esetében a gondolkodást és az intelligenciát, sok kutatás pedig egyetért abban, hogy a Turing-teszt kevésbé alkalmas arra, hogy megmérje egy számítógép intelligenciáját. A kritikák között olyan érvek szerepelnek, hogy a párbeszéd szimulálása az intelligenciát csak egy szűk szemszögből méri, vagy hogy attól még lehet intelligens egy gép, hogy nem képes emberi módon kommunikálni. Meg kell jegyezni azt is, hogy az emberek közül se teljesítené mindenki sikerrel a Turing tesztet (pl. kisgyerekek, vagy fogyatékkal élők), valamint azt is, hogy a teszten olyan ember is megbukhat, aki nem hajlandó a feltételek szerint együttműködni, ami természetesen nem az értelem hiányát jelenti. Néhány tanulmány a témában: (Copeland, 2000; Damassino, 2020; Neufeld és Finnstad, 2020).

Levesque et al. (2012) vezette be a Winograd-sémákat éppen arra a célra, hogy a segítségükkel tesztelni lehessen a gépi intelligenciát. A Winograd-sémáknak az alábbi feltételeknek kell megfelelniük:

1. a humán válaszadó számára könnyen egyértelműsíthető
2. a feloldás során nem lehet szelekciós megszorításokra támaszkodni
3. a feloldásban nem segít pl. a Google keresője

Ennek a kihívásnak az ereje az egyszerűségében rejlik: a sémákra bináris választ kell adni. Emellett az eredmény könnyen értelmezhető. Maguk a sémák a humán válaszadó számára nem okoznak nehézséget, ugyanakkor meglehetősen nehéz feladatot jelentenek egy gépi feloldó számára. Ennek az az oka, hogy a feloldáshoz szükséges világismeret és következtetési képesség az ember rendelkezésére áll, a számítógép számára viszont ezek nehezebben elérhető ismeretek.

A fent ismertetett három feltétel közül az első könnyen tesztelhető humán annotátorok bevonásával, a másik kettő viszont nehezebben tartható. Először is, a Winograd-sémák több nyelven is elérhetőek az interneten, tehát minden egyes séma esetében maga a megoldás megtalálható pl. a Google kereső segítségével. A szelekciós megszorításokkal kapcsolatban pedig meg kell jegyezni, hogy kimutatták, hogy valójában az eredeti sémák közül sok megoldható asszociáció segítségével. Az utóbbi problémára még részletesebben kitérek az 5.3.2 alfejezetben. A Winograd-sémák aktívan foglalkoztatják a nyelvtechnológiai közösséget, a téma legfrissebb eredményeit Kocijana et al. (2023) foglalja össze.

5.1. A Winograd-sémák

5.1.2. Benchmark adatbázisok Winograd-sémákkal

A gépi fordítás és az NLP területén a *benchmark* adatbázisok egy vagy több adatbázisból, és a hozzájuk tartozó metrikákból és kiértékelési módszerekből állnak. A benchmarkok tehát összehasonlíthatóvá teszik a velük kiértékelt eszközöket bizonyos szempontok alapján, amely szempontokban egyetértenek a kutatási terület közösségének tagjai. Ez az oka annak, hogy a benchmark adatbázisok korábbi *shared* taskok anyagából készültek (pl. GLUE (Wang et al., 2018), XTREME (Hu et al., 2020)), vagy a közösség igényei alapján készítették őket (pl. SuperGlue (Wang et al., 2020a), BIG-Bench (Ghazal et al., 2017)).

A General Language Understanding Evaluation (GLUE) benchmark adatbázist 2019-ben mutatták be. Olyan korpuszokból áll, amelyek különböző doménekből származó szövegeket tartalmaznak és különböző nehézségű nyelvmegértési tesztekre alkalmasak. A korpuszok már korábban is léteztek, esetleg kisebb módosításokat kellett rajtuk végezni.

A Winograd-sémák egy következtetési feladatként, más néven inferenciafeladatként (*natural language inference*, NLI) kerültek a gyűjteménybe. Az eredeti sémákat mondatpárookra alakították, majd a sémában szereplő mondatban a többértelmű névmást helyettesítették az egyik lehetséges antecedenssel. A feladat az, hogy megállapítsuk, hogy a második mondat következik-e az elsőből (amiben az egyik antecedens-jelölttel behelyettesített mondat van). A '0' címke azt jelenti, hogy a második mondat (amelyikben a névmást behelyettesítették az egyik antecedensjelölttel) nem következik az elsőből, az '1' címke pedig azt, hogy igen. A 9. példában a második mondat nem következik az elsőből, ezért kapott '0' címkét.

(9) **sentence1:**The drain is clogged with hair. It has to be cleaned.

sentence2:The hair has to be cleaned.

label: 0

A GLUE benchmarkban szereplő WNLI adatbázis 849 mondatpárból áll (634 mondatpár van a tanítóhalmazban, 70 mondatpár a fejlesztőhalmazban és 145 mondatpár a teszhalmazban). Ez a szám nagyobb, mintha vennénk az eredeti Winograd-sémákat és abból képeznénk a WNLI feladatokat. A 150 eredeti Winograd-sémából összesen 600 NLI-típusú feladat képezhető.

Egy másik benchmark gyűjtemény, a SuperGLUE elkészítésének motivációja az volt, hogy a GLUE tesztszei túl könnyűnek bizonyultak a nyelvmodellek számára. A SuperGLUE szintén tartalmazza a Winograd-sémákat, szintén bináris klasszifikációs feladatként, ahol minden példa egy mon-

5.2. A Winograd-sémák fordításai

datból áll, ahol egy névmás és egy főnév meg van jelölve, és az a feladat, hogy megmondjuk, hogy a névmásnak a megjelölt főnév-e az antecedense (lásd a 10. példát).

(10) The large ball crashed right through **the table** because **it** was made of styrofoam.

A GLUE benchmark WNLI korpuszán az eddigi legjobb, tehát a *state-of-the-art* (SOTA) eredményt jelenleg a Vega v1 Wang et al. (2020b) tartja (abszolút pontosság (accuracy): 97,9%). A SuperGLUE WSC korpuszán a SOTA eredményen ERNIE 3.0 (Sun et al., 2021) és a Microsoft Turing model (T-NLRv5) osztoznak (abszolút pontosság (accuracy): 97,3%). A humán eredmény mindkét feladatban 100%.

5.2. A Winograd-sémák fordításai

Az eredeti angol adatbázist több nyelvre is lefordították. A Winograd-sémák honlapja² szerencsére sorra veszi az eddig elkészült fordításokat. A fordítások között feltünteteti a japánt is (két verzióban: angol személynevekkel és japán személynevekkel), de további információkat nem, nem derül ki a fordítási módszer, nincs hozzá hivatkozás, egyedül a fordítók neve van megadva. Feltüntettek egy héber fordítást is, viszont sajnos a link nem működik.³ Egy 12 sémából álló kínai fordítás is szerepel a felsorolásban. Ezek mellett az orosz SuperGLUE (Shavrina et al., 2020) is tartalmaz Winograd-sémákat, a készítőik azonban nem adtak közre dokumentációt a fordítási módszerről. A következőkben röviden bemutatom az alaposabban dokumentált és publikált fordításokat.

5.2.1. Francia

Amsili és Seminck (2017) franciára fordította az angol Winograd-sémákat. Az adatbázis 107 sémát tartalmaz. A készítőik alaposan leírták a fordítási folyamat lépéseit és a nehézségeket is, amelyekkel szembekerültek. Az egyik fontos szempont, amire a franciára fordítás során figyelni kellett az az, hogy a névmás két lehetséges antecedensének egyeznie kell számban és nemben. A 9. példát csak módosításokkal fordíthatjuk franciára, mert a *haj* franciául (*cheveux*) többes számban szerepelne, amíg a *lefolyó* (*siphon*) egyes számban. Ennek megoldására a francia példában a *haj* helyett a *szappan* (*savon*) szó szerepel.

²<https://cs.nyu.edu/~davise/papers/WinogradSchemas/WS.html>

³A lekérdezés dátuma: 2022.03.28.

5.2. A Winograd-sémák fordításai

Egy másik probléma, hogy néhány séma esetében, amint a 11. példában a franciára fordított kifejezés önmagában kétértelmű lehet. Az *indiscreet* francia megfelelője ‘*indiscrete*’, és ‘*une personne indiscrete*’ jelentheti, hogy valaki kíváncsiskodó, amit angolul a *nosy* szóval jellemeznénk. Ebben az esetben az ‘*indiscrete*’ helyett inkább a *talkative* ‘*bavarde*’ melléknevet választották. Érdeemes megfigyelni, hogy a magyar esetében pontosan ugyanez a helyzet az *indiszkret* szóval.

(11) Susan knows all about Ann’s personal problems because she is [nosy /indiscreet].

Azokat a mondatpárokat, amelyeket nem tudtak úgy módosítani, hogy megfeleljenek az elvárásoknak, kivették az adatbázisból.

5.2.2. Portugál

de Melo et al. (2019) braziliai portugálra fordította az angol eredeti gyűjteményt. Három anyanyelvi beszélő dolgozott a mondatok fordításán. Minden mondatot az egyikük fordította, majd a másik kettő validálta a fordítást. Nyolc sémát nem tudtak lefordítani, így ezeket kihagyták az adatbázisból. Néhány sémát kicsit módosítani kellett hasonló okokból, mint a francia adatbázis esetén. Néhány sémát csak komolyabb módosítások árán tudtak lefordítani.

(12) The trophy doesn’t fit into the brown suitcase because it is too large.

A 12. példa portugál megfelelőjében a névmás ‘*ele*’ (*it*) antecedense könnyen feloldható, mivel egy hímnemű tárgyra utal, a mondatban pedig csak egyetlen hímnemű tárgy van, a *troféu* ‘*trophy*’ (a ‘*maleta*’ *suitcase* nőnemű). Ezt a sémát hasonlóképpen módosították, mint egyes francia sémák esetében, tehát két, nyelvtani nemből egyező lehetséges antecedens választottak a mondatokba. Ebben az esetben *trófea* *troféu* helyett a nőnemű ‘*medalha*’ *medal* főnevet választották, így a névmás is nőnemű párjára módosult (‘*ela*’).

A portugál Winograd-sémákat két változatban publikálták. Az egyik változat az eredeti angol tulajdonneveket tartalmazza, a másik változatban pedig az ismert emberek (pl. Shakespeare, Madonna) nevein kívül portugál személyneveket választottak.

5.3. További adatbázisok

5.2.3. Kínai

A fent ismertetett fordításokhoz hasonlóan a kínai változat, a Mandarinograd (Bernard és Han, 2020) készítői is szembesültek néhány problémával. Például az *I couldn't put the pot on the shelf because it was too [high/tall]* sémában a *tall* és a *high* melléknevek kínai megfelelője a 高. Ezért a *tall* helyett inkább a *short* melléknév kínai megfelelőjét használták.

5.3. További adatbázisok

További adatbázisok is léteznek, amelyeket a Winograd-sémák mintájára hoztak létre, tehát szintén mondatpárokat tartalmaznak és a többértelmű névmások antecedenskeresésének feladatára készültek. Ezek az adatbázisok gyakran próbálnak választ adni az eredeti Winograd-sémákat ért kritikákra.

5.3.1. Definite Pronoun Resolution Dataset

Rahman és Ng (2012) bevezette az ún. nehéz névmások (*difficult pronouns*) fogalmát. A nehéz névmások azokban a komplex esetekben tűnnek fel, amikor a mondat két mellékmondatot tartalmaz, amelyek egy kötőszóval vannak elválasztva egymástól, és két lehetséges antecedensjelölt van az első mellékmondatban és egy definit névmás a másodikban. A névmás és az antecedensjelöltek egyeznek egymással számban, személyben, nyelvtani nemben, emellett pedig közös szemantikai osztályba is tartoznak. Rahman és Ng (2012) példája (13) éppen ilyen mondatpár.

- (13) a. Ed shouted at Tim because he crashed the car.
b. Ed shouted at Tim because he was angry.

Ezekben a mondatokban a *he* névmás feloldása az ember számára nem nehéz feladat. Ennek az oka az, hogy az ember birtokában van annak a világismeretnek, hogy a mérges ember rákiabálhat másokra, valamint annak is is, hogy ha valaki hibát követ el, arra lehet hogy valaki mérges lesz, ezen kívül még annak is, hogy a karambol az egy ilyen jellegű hiba. Ezzel a tudással felvértezve az ember könnyen következtet arra, hogy Tim karambolozott és Ed volt mérges.

Megfigyeléseik alapján az egyébként jó eredményeket produkáló SOTA anaforafeloldók sem tudják a fenti példához hasonló nehéz névmásokat feloldani. A szabályalapú megoldások, amelyek gyakran a szintaxis kötésekre támaszkodnak szintén nem tudnak mit kezdeni ezekkel, hiszen a

5.3. További adatbázisok

névmás és a jelöltek külön mellékmondatban szerepelnek. A névmás grammatikai tulajdonságai szintén nem segítenek, mert két antecedensjelölttel is egyeznek. A hagyományos gépi tanuláson alapuló megoldások sem tudnak megbirkózni a feladattal, hiszen a tanítópéldákban felfedezhető felszíni jegyek nem fognak segíteni az eddig még nem látott példák esetében. A nehéz névmások antecedenskeresésének megoldására nem elég tehát a nyelvi jelenség alapos és részletes leírása szabályok formájában, vagy rengeteg tanítópélda biztosítása, az esetükben feltétlenül szükség van a nyelv és a szöveg mélyebb megértésére. Ez az a cél, amit az utóbbi években nagyon elterjedt neurális technikák próbálnak elérni. Ez a módszer lehetővé teszi például, hogy a nyelv mélyebb megértésével rendelkező modellt specifikusabb feladatok elvégzésére finomhangoljunk, amihez Definite Pronoun Resolution Datasethez hasonló tanítókorpuszokat használhatunk.

Fontos szem előtt tartani, hogy nem biztos, hogy az anaforafeloldó megoldások – legyen szó szabályalapú, statisztikai vagy neurális megközelítésűről – egyáltalán célozzák-e a nehéz névmások feloldását. A nehéz névmások nagyon ritkák a standard kiértékelő korpuszokban (pl. a Message Understanding Conference (MUC) 1995-ös és 1998-as shared taskok, valamint az ACE (Doddington et al., 2004) és az OntoNotes (Pradhan et al., 2007a) korpuszaiban is). Ahelyett, hogy a nehéz, de ritka eseteket fednék le, gyakran kifizetődőbb csak a gyakori, de könnyebben megoldható problémákra koncentrálni, hiszen így is jó eredmények érhetők el.

A standard kiértékelő korpuszok fent említett problémájára reagálva a Definite Pronoun Resolution Datasetben minden sémában egy kemény dió található, mivel mindegyik mondatban egy nehéz névmás szerepel. Az adatbázis 943 kézzel készített és annotált mondatpárt tartalmaz, közülük mindegyik egy nehéz névmást. Az adatbázis használható tanító- és kiértékelőadatként is egy anaforafeloldó számára, hiszen az adatot 70–30% arányban felosztották tanító és kiértékelő halmazra. Az adatbázis szabadon hozzáférhető⁴.

5.3.2. Winogrande

Sakaguchi et al. (2019) szerint a Winograd-sémák megbízhatósága megkérdőjelezhető, hiszen a jelenlegi neurális megoldások gyanúsán jól teljesítenek ebben a nehéz feladatban. Azt feltételezik, hogy a nagy neurális modellek rátanulhatnak bizonyos nem kívánt torzításokra (*bias*) az adatkészletben, annak ellenére, hogy a Winograd-sémák két fontos kritériuma éppen az ehhez hasonló torzítások elkerülésére irányul (ne lehessen egy egyszerű Google-kereséssel vagy szemantikai szelekciós megszorítások segítségével megoldani). Ez azt jelenti, hogy ezek a neurális modellek nagyon

⁴<https://www.hlt.utdallas.edu/~vince/data/emnlp12/>

5.3. További adatbázisok

jól teljesítenek, mert megoldják ugyan a problémát, de nem azért, mert olyan ügyesek lennének az anaforafeloldásban. A probléma hangsúlyozására azt mutatták ki, hogy az eredeti Winograd-sémák 13,5%-a asszociatív. A 14b. példába esetében belátható, hogy általában az épületek híresek, és nem a térképek, ugyanakkor a 14a. példában nincs jelen ez a hatás. Nem nehéz belátni, hogy egy adathalmaz nem a legmegfelelőbb egy anaforafeloldási rendszer vagy egy nyelvi modell értékelésére, ha minden hetedik-nyolcadik mondat esetében van egy kiskapu a valódi probléma megkerülésére.

- (14) a. Bill passed the gameboy to John because *his* turn was over.
 b. I'm sure that my map will show this **building**; *it* is very **famous**.

Sakaguchi et al. (2019) megoldása a problémára, hogy ki kell venni a feladatot a humán annotátorok kezéből, mivel az emberek nehezen tudnak teljesen elfogulatlanul sémát alkotni anélkül, hogy az véletlenül nemkívánatos torzításokat tartalmazzon. Gyűjteményük⁵ 44 000 problémát tartalmaz. Az adatot egy gondosan megtervezett crowdsourcing projektben építették és egy szűrőalgoritmust alkalmaztak a torzítások eltávolítására. Míg az ember számára triviális marad a sémák megoldása (94%-os pontosság), addig a Winogrande már tényleg feladja a leckét a számítógépnek, beszámolójuk szerint a legjobb eredmények 15-35%-kal elmaradnak az emberi teljesítménytől.

5.3.3. Wino-X

A Wino-X (Emelin és Sennrich, 2021) Winograd-sémákat tartalmaz angol, német, francia és orosz nyelveken. Az erőforrást kétféle célra lehet használni. Egyrészt a segítségével vizsgálható, hogy a gépi fordítás alkalmas-e koreferenciafeloldásra olyan szövegekben, ahol a feladat csak világismeret és következtetés segítségével oldható meg, másrészt pedig a többnyelvű modellek következtetési képességeinek tesztelésére. A Wino-X mindkét alkorpusza a Winogrande korpuszából tartalmaz mondatokat és azok fordításait. A névmások nemének fordításából adódó problémák elkerülése érdekében csak az *it* névmással hivatkozott élettelen antecedenssekkel rendelkező mondatokat használták fel. A Wino-X két részhalmaza – az MT-Wino-X a neurális gépi fordítási modellek kiértékeléséhez és az LM-Wino-X a többnyelvű nyelvi modellekhez – formátumában különbözik. Az előbbi a feladatot fordítási teszt formájában tartalmazza (15. példa), míg az utóbbi a Winogrande hiánypótló formátumát veszi át (16. példa).

⁵<https://github.com/allenai/winogrande>

5.3. További adatbázisok

- (15) **forrásmondat:** I dusted **the dresser** in the bedroom with **a rag** until *it* was free of dust.
helyes fordítás: Ich staubte **die Kommode** im Schlafzimmer mit einem Lappen ab, bis *sie* staubfrei war.
helytelen fordítás: Ich staubte die Kommode im Schlafzimmer mit **einem Lappen** ab, bis *er* staubfrei war.
- (16) **angol kontextus:** Adam chose to sleep on **a sofa** instead of **a bed** because _ was much more comfortable.
helyes filler: the sofa
helytelen filler: the bed
német kontextus: Adam entschied sich dafür, auf **einem Sofa** statt auf **einem Bett** zu schlafen, weil _ viel bequemer war.
helyes filler: das Sofa
helytelen filler: das Bett

5.3.4. XWINO

Az XWINO (Tikhonov és Ryabinin, 2021) hat nyelven tartalmaz Winograd-sémákat és azokhoz hasonló feladatokat. Az erőforrás így alkalmas a többnyelvű modellek következtetési képességeinek tesztelésére. Az angol nyelvből az eredeti Winograd-sémákat tartalmazza, valamint a SuperGLUE benchmarkot és a Definite Pronoun Resolution Dataset anyagát. Portugálból, franciából, oroszból és japánból az 5.2. fejezetben ismertetett fordításokat használták. A Mandarinograd adatát az eltérő formátuma miatt kihagyták. Az adatbázis⁶ 3 961 sémát tartalmaz.

5.3.5. Pronoun Disambiguation Problems (PDP)

Az anaforafelolás feladatát megoldó eszközök teljesítményét nem csak a Winograd-sémákon lehet ki-mérni. Eredetileg a 2016-os Winograd-séma Kihívás (Winograd Schema Challenge) beugró feladata volt a PDP (Pronoun Disambiguation Problems)⁷ (Morgenstern et al., 2016; Davis et al., 2017) halma-

⁶https://github.com/yandex-research/crosslingual_winograd

⁷<https://commonsensereasoning.org/disambiguation.html>

5.4. A magyar Winograd-sémák és hasonló adatbázisok

za. A PDP 122 rövid szövegből áll. A szövegrészeket regényekből származnak. A mondatokat úgy módosították, hogy a kontextus és a háttértörténet ismert legyen. Emellett az egyes főnevek vagy névmások számát vagy nyelvtani nemét is módosítani kellett, hogy fennálljon a többértelműség. A karakterek neveit is megváltoztatták. Az egyes szövegrészek több célnévmást is tartalmazhatnak.

A Winograd-sémákhoz hasonlóan a PDP szövegekben szereplő névmások antecedensének megkereséséhez szükséges a világismeret és a következtetés, amint a 17. példa is mutatja.

- (17) **mondat:** Always before, Larry had helped Dad with his work. But he could not help him now, for Dad said that his boss at the railroad company would not want anyone but him to work in the office.
- részlet:** he could not help
- A válasz:** Larry
- B válasz:** Dad
- helyes válasz:** A

A PDP szövegek két formátumban is elérhetők, ám ez a két forrás nem ugyanazokat a mondatokat tartalmazza. A weboldalon 62 szöveg van, egy további link alatt pedig újabb 60 szöveg. Az utóbbi teljes egészében része a GLUE benchmark WNLI adatbázisnak (lásd az 5.1.2), valamint szerepel benne további egy példa a honlapon szereplő gyűjteményből.

A Winograd-sémákkal és az azok mintájára készült adatbázisokkal összevetve a PDP gyűjteménye nem mondatpárokból áll, a rövid szövegek kissé közelebb állnak a valódi szövegekhez és kevésbé mesterségesek annak ellenére, hogy ezek is szerkesztett szövegek. Egy olyan erőforrás elkészítése, ami valódi, szerkesztetlen szövegekből áll, nagyon nagy kihívás lenne, hiszen a szerkesztetlen szövegek előkészítése, elemzése és annotálása nagyon nehéz feladat.

5.4. A magyar Winograd-sémák és hasonló adatbázisok

Ebben az alfejezetben ismertetem a társszerzővel készített erőforrásokat. Az összes adatbázis az angol megfelelőjének fordításával készült, amihez gépi fordítót és humán validációt alkalmaztunk.

5.4. A magyar Winograd-sémák és hasonló adatbázisok

5.4.1. HuWS: magyar Winograd-sémák

A Winograd-sémák esetében a fordítást ketten validáltuk, mindketten nyelvészek. Bizonyos sémákat kizártunk, mert nem tudtuk őket úgy magyarra fordítani, hogy megfeleljenek a Winograd-séma elvárásainak. A 18. példa esetében nem tudtuk magyarra fordítani a *breaking her silence, breaking her concentration* frázisokat úgy, hogy a mondatpár tagjai csak egy frázisban különbözzenek és megmaradjon a birtokos szerkezet.

(18) Lily spoke to Donna, breaking her [silence/concentration].

A 11. példa már a francia fordítással kapcsolatban is felmerült. Az *indiscreet* és a *nosy* fordítása a magyarban ugyanaz a szó: 'indiszkrét'. Néhány séma pedig bizonyos módosítások árán volt magyarra fordítható.

A fordításkor azt is figyelembe kellett venni, hogy a névmások eltérően viselkedhetnek az egyes nyelvekben. A magyarban mint pro-drop nyelvben bizonyos esetekben a mondatból kimaradhat a testes névmás. A névmás elhagyása nem kötelező, de sok esetben jelöltebb mondatot eredményez, ha testesen megjelenik a névmás. A fordítás során figyeltünk rá, hogy a mondatpárok mindkét mondatában egységesen kezeljük ezeket az eseteket, tehát vagy mindkettőben elhagyott, vagy mindkettőben testes névmás jelenjen meg. Pléh és Radics (1976) alapján ha két mondat közül a második mondat alanyi pozíciójában mutatónévmás szerepel, akkor az alanyváltásra utal. Ez azt jelenti, hogy az ilyen mondatokban ha csak két lehetséges antecedens szerepel az első mondatban és közülük egyik az alany, akkor nem áll fenn az a többértelműség, ami a Winograd-sémákban szereplő mondatok sajátja. A 19a. példában egy ilyen mondatpár látható. Éppen ezért sok Winograd-sémát csak elhagyott névmásokkal volt lehetséges magyarra fordítani, mint a 19b. példában szerepel.

(19) a. A férfi nem tudta felemelni a fiát, mert az olyan [*gyenge/nehéz] volt.

b. A férfi nem tudta felemelni a fiát, mert olyan [gyenge/nehéz] volt.

Ennek az ellenkezője is előfordul azonban, amikor csak testes névmásokkal lehet fenntartani a szerkezeti többértelműséget. Ha például a 20. példában látható mondatok második tagmondatában elhagyott alany szerepelne, akkor nem állna fenn a többértelműség, mert akkor csak az első mondat alanya lehetne az antecedense.

5.4. A magyar Winograd-sémák és hasonló adatbázisok

(20) A tűzoltók a rendőrök [után/előtt] érkeztek ki, mert ők olyan messziről jöttek.

Az angol személyneveket lefordítottuk magyarra, kivéve a híres emberek neveit. Az eredeti gyűjteményből csak 122 mondatpárt sikerült magyarra fordítanunk. A magyar Winograd-sémák szabadon hozzáférhetők⁸, json formátumban használhatók.

5.4.2. HuWNLI: A magyar WNLI adatbázis

A GLUE gyakorlatát követve inferenciaadatbázissá alakítottuk át a magyarra fordított Winograd-sémákat, hogy megfelelő tanítóanyagot biztosítsunk a neurális modellek számára. Egy sémából (21. példa) négy mondatot és a hozzájuk tartozó állítást hoztunk létre. A kétértelmű névmást behelyettesítettük a mondatpár mindkét tagjában a két lehetséges antecedenssel, így jött létre a 22. példában látható négy mondat. Ilyen módon egy bináris mondatklasszifikációs feladattá alakítottuk a sémákat, ahol az a feladat, hogy megjósoljuk, hogy a második mondat — ahol a névmás helyett egy lehetséges antecedense ismétlődik meg – következik-e az elsőből vagy nem. Ennek megfelelően kétféle címke járulhat az egyes mondatokhoz és a hozzájuk tartozó állításhoz kétféle címke járulhat: *entailment* és *non-entailment*.

(21) A trófea nem fér bele a barna bőröndbe, mert túl [nagy/kicsi].

(22) a. A trófea nem fér bele a barna bőröndbe, mert túl nagy. A trófea túl nagy.

címke: entailment

b. A trófea nem fér bele a barna bőröndbe, mert túl nagy. A bőrönd túl nagy.

címke: not-entailment

c. A trófea nem fér bele a barna bőröndbe, mert túl kicsi. A trófea túl kicsi.

címke: not-entailment

d. A trófea nem fér bele a barna bőröndbe, mert túl kicsi. A bőrönd túl kicsi.

címke: entailment

A WNLI adatbázis három részre oszlik: tanító, fejlesztő és tesztelő halmazra. A WNLI-ben szereplő mondatok az eredeti Winograd-sémák mellett a PDP-ből tartalmaznak mondatokat. A magyar WNLI adatbázis tehát az angol WNLI magyarra fordított változata, és a tanító-, fejlesztő- és tesztal-

⁸<https://github.com/nytud/HuWS>

5.4. A magyar Winograd-sémák és hasonló adatbázisok

mazra bontásnál is az eredetit követtük, 596 mondatpár tartozik a tanítóadatba, 52 a fejlesztő adatba és 134 a teszhalmazba. Ezek a számok nem egyeznek meg pontosan az eredeti WNLI adatbázis méreteivel, mert bizonyos mondatok nem voltak magyarra fordíthatók. Két hibás címkét is találtunk az eredeti WNLI-ben (ID 347 és ID 464), ezeket kijavítottuk a magyar adatban. Néhány az eredeti Winograd-sémák közül hiányzott az eredeti WNLI-ből (pl. 22-es és 29-es sorszámú), ezeket a magyar változat tanítóanyagába tettük. Mivel az eredeti WNLI teszhalmazának a címkéi nem nyilvánosak, ezért humán annotátorok címkézték meg a magyar teszhalmazt. A HuWNLI a HuLU (Hungarian Language Understanding Benchmark Kit) (Ligeti-Nagy et al., 2022) része, de önálló GitHub repozitóriumban⁹ is elérhető, json formátumban.

5.4.3. PWS: Parallel Winograd-sémák

Amint az 5.2. fejezetben már ismertettem, az eredeti angol nyelvű Winograd-sémákat több nyelvre is lefordították. Az összes elérhető fordítást összegyűjtöttük és egy párhuzamos korpuszt, a PWS-t (Parallel Winograd Schemata) készítettünk belőlük. A párhuzamos korpusz megfelelő terepet biztosít arra, hogy megvizsgálhassuk a névmások viselkedését a korpuszban szereplő nyelvek esetében. A PWS a következő fordításokat tartalmazza az angol eredeti mellett: portugál, francia, kínai, japán, orosz és magyar. A portugál és a japán mindkét verziójában megjelenik a korpuszban: egyrészt az angol eredeti személynevekkel, másrészt pedig a célnyelvi személynevekkel.

Az erőforrásokban fellelhető összes metaadatot és információt (fordító neve, megjegyzés a fordítási nehézségekről stb.) megőriztük és feltüntettük a PWS-ben. A PWS egy táblázatban kapott helyet, ahol minden mondathoz tartozik egy sor, az oszlopokban pedig az egyes fordítások (a mondatok, a kérdések vagy mondatrészletek (*snippet*), a válaszlehetőségek és a helyes válasz fordításai) szerepelnek. Nem minden angol eredeti volt fordítható az egyes nyelvekre, így a párhuzamos korpusz bizonyos mezői üresen maradtak.

Amint az 5.4.1. fejezetben már kiemeltem, hogy bizonyos sémák magyarra fordításakor módosításokat kellett tenni az angol eredetin. Ugyanez igaz a többi nyelvre fordítás esetében is, ezért a mondatok nem feltétlenül tekinthetők egymás pontos fordításainak. A PWS-nek ezt a tulajdonságát érdemes szem előtt tartani. A PWS adatbázis szabadon hozzáférhető¹⁰.

⁹<https://github.com/nytud/HuWNLI>

¹⁰<https://github.com/nytud/PWS>

5.4. A magyar Winograd-sémák és hasonló adatbázisok

5.4.4. HAPP: Hungarian Ambiguous Pronoun Problems

A Definite Pronoun Resolution Dataset (Rahman és Ng, 2012) 1 882 mondatát lefordítottuk magyarra. A magyar fordításban minden mondatpárra jellemző, hogy vagy mindkettő esetében elhagyott célnév más van, vagy mindkettő esetében testes.

Néhány kötőszó tökéletesen működik az angol esetében, de másként viselkednek a magyarban. A 23. példa esetében nem tudjuk úgy lefordítani a mondatpárt, hogy mindkettőben testes legyen a névmás, vagy mindkettőben elhagyott. Elsősorban azokban a mondatokban jelentkezik a probléma, ahol az 'és' vagy a 'majd' kötőszók kötik össze a két mondatot és a célnév más alanyi szerepben áll.

- (23) a. A madár beleevett a pitébe, és [meghalt/*tönkrement].
 b. A madár beleevett a pitébe, és az [*meghalt/tönkrement].

A megoldásunk erre a problémára az volt, hogy két mondatba szedtük szét ezeket az eseteket, amint a 24. példában látható. Az eredeti angol gyűjtemény egyébként 134 *and* kötőszót tartalmaz.

- (24) A madár beleevett a pitébe. [Meghalt/Tönkrement].

Ha a célnév más nem alanyi szerepben áll a második mondatban, amint a 25. példában látható, akkor a mondat lefordítható a fenti nehézség nélkül.

- (25) A labda eltalálta az ablakot, és Vili [megjavította/elkapta].

Egyéb kötőszavak mint a 'mert', 'pedig', 'de', 'hogy', 'így' nem okoznak ilyen problémát.

A legtöbb esetben a névmásnak egyeznie kell számban az antecedensével. Vannak kivételek is, például amikor az antecedens egy gyűjtőnév, ekkor – mint a 26. példában is látható – a névmás állhat többes számban is. Itt a második mondat többes számú alanya – egy elhagyott névmás képében – visszautalhat az első mondatban szereplő többes számú tárgyra is, de az egyes számban álló gyűjtőnévvel kifejezett alanyra is, a második mondat igéjének függvényében.

5.5. Összefoglalás

(26) A rendőrség letartóztatta a lázadókat, mert [megelőzték a bajt/bajt okoztak].

Az eredeti adatbázis 30/70% arányban volt felosztva tanító és tesztelő halmazra, ezt a felosztást megtartottuk a magyar fordítás esetében is. A HAPP adatbázis szabadon hozzáférhető¹¹, az adatbázis json formátumban használható. A 10. fejezet egy olyan kísérletet ismertet, amelyben a HAPP adatbázisát használtam ahhoz, hogy megvizsgáljam, milyen jól teljesít a ChatGPT a többértelmű névmási anaforák feloldásának feladatában.

5.5. Összefoglalás

A nagy nyelvmodellek elterjedésének köszönhetően a benchmark adatbázisok készítése a nyelvtechnológia aktuális feladata, így ezeknek a magyar sémagyűjteményeknek az elkészítésével hozzájárulhattunk ahhoz, hogy magyar adaton is vizsgálhatóak legyenek a nagy nyelvmodellek nyelvértési képességei. A 10. fejezetben egy olyan kísérletet ismertetek, amelyben az itt ismertetett magyar HAPP gyűjteményt használtam fel arra, hogy megvizsgáljam, hogy a ChatGPT hogyan teljesít a többértelmű névmási anaforák feloldásában. A különböző nyelvekre elkészített adatbázisok emellett lehetővé teszik, hogy a többértelmű névmási anaforák viselkedését nyelvenként összevessük.

¹¹<https://github.com/nytud/HAPP>

II. rész

Eszközök

6. fejezet

Anafora- és koreferenciafeloldó eszközök

A disszertáció második fele eszközöket kínál az anafora- és koreferenciafeloldás feladatának megoldására. Az általam készített eszközök bemutatása előtt ismertetek néhány anafora- és koreferenciafeloldó módszert, valamint részletesebben kitérek a magyar előzményekre a témában. A módszerek ismertetését [Hirst \(1981\)](#), [Mitkov \(1999\)](#) és [Ng \(2017\)](#) alapján teszem meg.

Az anafora- és koreferenciafeloldás feladatának megoldására készültek szabályalapú, felügyelt gépi tanulást vagy neurális hálót alkalmazó, valamint hibrid módszerek is. A szabályalapú megoldások esetében a nyelvtechnológus által definiált szabályok elsősorban a szintaktikai szerkezetre, nyelvtani és lexikális tulajdonságokra (pl. egyezés számban és személyben, nyelvtani nemben), valamint a mondat lineáris szerkezetére (közelség) támaszkodnak. A nyelvtechnológusnak nagy a kontrollja a rendszer fölött, valamint a szabályok implementálása is egyszerű, ráadásul a kimenet is könnyen értelmezhető. A szabályalapú módszerek gyengesége a törékenysége: minél bonyolultabbak a szövegben található nyelvi jelenségek és minél több és összetettebb szabály van, úgy válik egyre nehezebben átláthatóvá maga a szabályrendszer is. Az anafora- és koreferenciafeloldás esetében ráadásul a szabályalapú megoldásoknál mindig szükség van további nyelvi (pl. szintaktikai, morfológiai) elemzésekre a bemenetben, amelyek minősége nagyban befolyásolhatja a feloldó eredményét is.

6.1. Módszerek az anafora- és a koreferenciafeloldásban

6.1. Módszerek az anafora- és a koreferenciafeloldásban

A szabályalapú megoldásra a legkorábbi példák között van [Bobrow \(1964\)](#) STUDENT programja, amely középiskolai szöveges matematikafeladatokat old meg. A programnak néhány nagyon egyszerű szabálya vonatkozik az anaforákra, azon belül elsősorban a variációkra, valamint a teljes és nem teljes ismétlésekre. A bemeneten nem történik hagyományos értelemben vett szintaktikai elemzés, csupán mintaillesztés zajlik a háttérben.

Egy másik korai példa [Winograd \(1972\)](#) SHRDLU programja, amelyben a felhasználó természetesnyelven megfogalmazott utasításokkal irányíthat egy virtuális térben, mely különböző alakú és méretű színes tárgyakból áll. A tárgyak mozgatása során párbeszéd zajlik a számítógép és a felhasználó között, a számítógépnek memóriája és a fizika törvényeivel kapcsolatos ismeretei vannak, a felhasználónak pedig lehetősége van elnevezni a tárgyakat, amelyekre a társalgás során referálni lehet. Érthető, hogy miért volt kiemelt jelentőségük a névmásoknak a SHRDLU-ban. A névmások kezelésére az algoritmus az addig előforduló főnévi csoportokat mint lehetséges antecedenseket kezeli. Minden lehetséges antecedensjelöltet számba vesz, mielőtt döntést hoz. A döntés során figyelembe veszi a szintaktikai szerkezetet és ez alapján sorrendezi a jelölteket.

Ez a két korai példa jól mutatja, hogy az anaforikus kapcsolatok kezelése a komplexebb nyelvtechnológiai feladatok szükséges alfeladata. A két problémában (matematikafeladat és tárgyak navigálása a térben) megkívánja, hogy a szövegbeli szereplők hivatkozhatók legyenek, ehhez pedig szükséges a koreferens elemek összekapcsolása a szövegben.

A további szabályalapú megoldásokban általában nagy szerepet kap a szintaktikai szerkezet. [Hobbs \(1978\)](#) algoritmus a szintaxisfákon alapuló szabályokat (pl. kötéselvek) veszi figyelembe, a szövegbeli elemek egyéb tulajdonságait (pl. jelentés) nem. Érdekes, hogy ehhez a megoldáshoz készült először korpuszalapú kiértékelés. A kiértékelőkorpusz különböző műfajú szövegekben 300 darab névmást tartalmazott. Nyilvánvaló korlátai ellenére a későbbiekben sokáig ez a rendszer szolgáltatta a baseline eredményt a későbbi megoldások számára.

A szemantikai információk szintén fontos szerepet játszottak már a korai megoldásokban is. [Wilks \(2007\)](#) elemző és generáló rendszerében a névmások kezelése kézzel kódolt szemantikai (lexikális) információk alapján történik. Ezek a jelentéssel kapcsolatos információk vonatkozhatnak pl. a főnevekkel jelölt dolog élőségére, fizikai tulajdonságaira stb.

A szabályalapú módszerek mellett megjelentek a hagyományos gépi tanulási módszerek is. A felügyelt gépi tanulást alkalmazó módszerek esetében nagy mennyiségű és jó minőségű tanító- és ki-

6.2. Magyar anafora- és koreferenciafeloldók

értékelőadatra, tehát kézzel annotált korpuszra van szükség. A kézzel annotált korpuszok előállítására erőforrásigényes és sok tervezést igénylő feladat, ám ha rendelkezésünkre áll a megfelelő mennyiségű és minőségű alapanyag, már csak a feladatra alkalmas gépi tanuló algoritmust kell kiválasztanunk, valamint a megfelelő jegyeket meghatározni a sikeres tanításhoz. Ahelyett, hogy kiemelnék egy-egy hagyományos gépi tanulást alkalmazó megoldást, igyekszem inkább az egyes módszerek főbb tulajdonságait megvilágítani. A gépi tanulóval készült rendszerek között találunk *mention-pair* modelleket, amelyek esetében bináris klasszifikáció történik, miszerint fennáll-e koreferenciakapcsolat az egyes utalás-párok tagjai között, vagy nem. A bináris klasszifikáció helyett alkalmazhatunk ún. *mention-ranking* modelleket is, amelyek a lehetséges jelölteket sorrendezik. Az ún. *entity-based* megoldások az egy koreferenciaklaszterbe tartozó elemek jegyeit igyekeznek figyelembe venni ahelyett, hogy csak az egyes utalások lokális tulajdonságait tekintenék. A további módszerek mellett fontos megemlíteni, hogy a szintaxis mellett a lexikális vagy szemantikai tulajdonságoknak is nagy szerepe van a koreferenciafeloldásban.

A neurális technikák az anafora- és koreferenciafeloldás területén is egyre elterjedtebbek. A rekurrens neurális hálók, valamint a transzformer alapú modellek jól megragadják a szövegekben előforduló komplex mintázatokat és a szöveg elemei közötti kapcsolatokat. A NeuralCoref¹ (Wolf, 2017) a SpaCy² (Honnibal, 2015) rendszer pipeline-jába illeszthető koreferenciafeloldó eszköz, amit neurális háló segítségével készítettek és angol nyelvű szövegekre működik. Joshi et al. (2019) pedig egy magas szintű, transzformer alapú koreferenciafeloldó rendszert készítettek, amelyben a *c2f-coref* modellt (Lee et al., 2018) fejlesztették tovább a BERT integrálásával.

6.2. Magyar anafora- és koreferenciafeloldók

A magyar nyelvre eddig több tudásalapú, szabályokkal operáló anafora- és koreferenciafeloldó megoldás is született. Lejtovicz és Kardkovács (2007) szabályalapú anaforafeloldója a BFP algoritmust (Brennan et al., 1987) implementálta. A BFP algoritmus a diskurzus modellezésére megalkotott ún. *centering* elméleten alapul. A centering elmélet azért alkalmas a névmási anaforafeloldás feladatának megoldására, mert a névmásokat a központi témára való figyelemirányítás eszközeként kezeli. A BFP algoritmust azért találták megfelelőnek a magyar szövegekre, mert a BFP-t a magyarhoz hasonló információsszerkezetű nyelvekre lett kifejlesztve. A cikk részletesen leírja a BFP algoritmus három lépését és azt, hogy ezeket a lépéseket hogyan módosították a magyar nyelv esetében. Egy

¹<https://github.com/huggingface/neuralcoref>

²<https://spacy.io/>

6.2. Magyar anafora- és koreferenciafeloldók

példa levezetése után pedig a Szeged Treebank (Csendes et al., 2005) 500 mondatán értékelték ki a megoldásukat. Az algoritmus 39,6%-os találati arányt ért el az összes anaforát beleszámítva. Sajnos a program nincs közzétéve, ezért az eredmény nem reprodukálható.

Miháltz et al. (2007) és Miháltz (2012) tudásalapú koreferenciafeloldója a mondatok mély szintaktikai szerkezete mellett a kötéselmélet tételeire és pszicholingvisztikai megállapításokra támaszkodik, valamint a Magyar WordNet (Miháltz et al., 2008) ontológiában tárolt nyelvi tudást is kiaknázza. Ez utóbbi a koreferenciakapcsolatok esetében segít, míg az előbbieket az anaforikus kapcsolatok felismerésében. Az algoritmus 4 lépést tartalmaz: előszűrés, az antecedensek listázása, a jelöltek szűrése majd az antecedens kiválasztása a fennmaradó jelöltek közül. A tulajdonnevek, a határozott névelős köznevek és a névmások – zérónévmások, személyes névmások és az alanyi szerepben álló ‘az’ mutatónévmások – antecedensét keresi a szövegben, az elemek antecedenskeresése pedig meghatározott sorrendben történik. A cikk csak részleges kiértékelést ismertet, amihez kézzel annotált kiértékelőkorpuszt alkalmaztak. A korpusz 5 darab, általános iskolai történelemkönyvből kiemelt szöveget tartalmaz, összesen 652 mondatnyit. Az algoritmus 69%-os átlagos pontosságot ért el a kiértékelőkorpuszon. A részletes hibaanalízisből kiderül, hogy az algoritmus sokkal jobb eredményt is elérhetne (83%), ha a koreferenciafeloldás bemeneteként olyan szöveget használnánk, aminek a szintaktikai elemzése kézzel ellenőrizve van.

Munkácsy és Farkas (2016) statisztikai alapú megoldást kínált a koreferenciafeloldás feladatára. A SzegedKoref (Vincze et al., 2018) anyagán tanították a HOTCoref (Roesiger és Kuhn, 2016) rendszert, majd a rendszer moduljait alakították át a magyar korpusznak megfelelően. Megoldásukat többféle metrikával is kiértékeltek, a kiértékeléshez a CoNLL-2011 Shared Task hivatalos kiértékelő szkriptjét³ használták a MUC, BCUC, CEAFM és CEAFE metrikákkal. A négy mérőszám átlaga 43,0525 lett, amit összevetettek egy ugyanekkora tanítóadattal feltanított angol rendszer eredményeivel (54,6975), igaz, sajnos nem közölték, hogy melyik angol rendszerről és milyen tanítóadatról van szó.

Tudásszegény megoldást a magyar névmási anaforák antecedenskeresésére Kovács (2021) kínál, amelyben gépi tanulási kísérleteket végez a különböző névmástípusok esetében. A megoldását a SzegedKoref és a KorKor anyagán külön is kiértékelte, valamint saját teszt példákat is konstruált, hogy minél több szempontból meg tudja becsülni a feloldó minőségét.

³<https://conll.cemantix.org/2011/software.html>

7. fejezet

Zérónévmás-beszúrás két módszerrel

A pro-drop nyelvek esetében a zérónévmás-beszúrás az anaforafeloldás szükséges alfeladatának tűnik, a szakirodalomban azonban nem találtam magyar megoldásokat. Ha az anaforafeloldást olyan információkinyerési feladat részeként tekintjük, ahol például azt szeretnénk feltárni, hogy a szövegben előforduló szereplők milyen cselekvéseket hajtottak végre, milyen események történtek velük, milyen állítások vonatkoznak rájuk, akkor feltétlenül szükséges a cselekvések, események, állítások testetlen bővítményeit is felfedni.

Ebben a fejezetben két megoldást is kínálok zérónévmások magyar szövegekbe történő beillesztésére: egy szabályalapút, amelyet a KorKor korpusz építéséhez fejlesztettem, és amely az `emt sv` keretrendszerében modulként is használható, valamint egy neurális megoldást, amelyet elemzetlen, nyers szövegeken lehet használni. A két zérónévmás-beszűrő ismertetése előtt részletesebben kitérek arra, hogy az általam készített zérónévmás-beszűrők mely zérónévmásokat célozzák és miért.

7.1. Zérónévmások és testetlen alanyok

A KorKor építésekor azt a célt tűztem ki, hogy a korpuszban megjelenő anafora- és koreferenciaannotációból kinyerhető legyen, hogy a szövegekben megjelenő szereplők milyen eseményekben vesznek részt, milyen cselekvéseket hajtanak végre, illetve milyen állítások vonatkoznak rájuk. Ahhoz, hogy az egyes szereplőkre vonatkozó állításokat, cselekvéseket, eseményeket ki tudjuk nyerni, mindenképpen szükség van a zérónévmások beillesztésére is. A zérónévmások is anaforikus kapcsolatokban

7.1. Zérónévmások és testetlen alanyok

részt vevő elemekként jelennek meg a korpuszban, így a korpuszépítés későbbi fázisában anaforaként és antecedensként annotáltuk őket.

Az itt ismertetett zérónévmás-beszűrő megoldások, így a KorKor építéséhez készített zérónévmás-beszűrő és az azon alapuló emZero, valamint a neurális zérónévmás-beszűrő az alábbi helyekre illeszti be a zérónévmásokat:

- 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
- és ragozott és ragozatlan infinitívusz alanyának.

A fenti felsorolás alapján két kérdés is felmerülhet. Kérdéses egyrészt az, hogy az infinitívusz testetlen vonzatai tekinthetők-e zérónévmásoknak, másrészt pedig felmerülhet, hogy érdemes lenne a további helyeken feltételezhető zérónévmásokat is kezelni.

Az elméleti nyelvészeti szakirodalom megkülönbözteti egymástól a zérónévmásokat és az ige-vek testetlen vonzatait (Kiefer, 1992; É. Kiss et al., 2003; É. Kiss, 2004). Az előbbieket gyakran *pro* jelöléssel ábrázolják, utóbbiakat pedig *PRO*-val. A különbség köztük az, hogy a zérónévmás (*pro*) száma és személye a vele egyező ige vagy birtokszó személyragjából következtethető ki, míg a *PRO* esetében ez nem lehetséges. Az infinitívuszok esetében a *PRO* antecedensét a ragozott ige határozza meg, az ún. alanyi kontroll esetében az infinitívusz alanya a ragozott ige alanyával (27a), a tárgyi kontroll esetében pedig a tárgyával egyezik meg (27b), valamint találunk néhány példát datívuszi kontrollra is (27c).

- (27) a. A cica_i nem akar [PRO elindulni].
 b. A cicát_i láttam [PRO_i vadászni].
 c. A cicának_i nem hagyom [PRO_i a madarakat megfogni].

Valóban, a zérónévmásokat és az infinitívuszok testetlen vonzatait a szintaxis szempontjából fontos elválasztani, a bevezetésben definiált feladat (a szövegben előforduló szereplőkre vonatkozó állítások, cselekvések, események felfedése) szempontjából azonban mégis rokon jelenségek. Az infinitívuszok vonzatai éppen a kontrollelmélet alapján ráadásul nagyon gyakran olyan szereplők, amelyeknek a főmondat valamely argumentumával azonos a referenciájuk. Ez alól az általános alanyú infinitívusz a kivétel, amint a 28. példában látható.

- (28) Fontos [PRO felkészülni a vizsgára].

7.1. Zérónévmások és testetlen alanyok

A 29. példa esetében a ragozott infinitívuszok alanya pedig szintén zérónévmás, amit az infinitívuszon megjelenő személyrag tesz lehetővé.

(29) Fontos [pro_i felkészülnöd_i a vizsgára].

Összefoglalva tehát a ragozatlan infinitívusz alanyát amiatt kezeltem a zérónévmásokhoz hasonló módon, mert a PRO és az őt kontrolláló összetevő között is koreferenciakapcsolat áll fenn, így ezek az alanyok is részt vehetnek az anaforikus kapcsolatokban. A KorKor korpuszban az alanyi, tárgyi és datívuszi kontroll az anaforikus kapcsolatok mintájára van tehát jelölve: az infinitívusz alanyául szolgáló zérónévmás az anafora, az antecedense pedig a ragozott ige alanya, tárgya, vagy datívuszi vonzata.

Zérónévmásokat nemcsak a fenti felsorolásban megjelölt helyeken feltételezhetünk, hanem további igevevek esetében is. [É. Kiss et al. \(2003\)](#) és [É. Kiss \(2004\)](#) szerint a módhatározói szerepű határozói igevevek esetében a határozói igenév alanyának referenciája a ragozott ige alanyáéval azonos, tehát ebben az esetben mindig alanyi kontroll áll fenn. Amint a 30a. példa esetében ez valóban így van, ám amint a 30b. és a 30c. példákban látható, mégis tudunk olyan példamondatot alkotni, amelyben nem alanyi kontroll áll fenn, ebben az esetben a főige nem alanyi vonzatának referenciájával azonos a PRO referenciája.

- (30) a. PRO Kilépve az utcára Mari rögtön fázni kezdett.
 b. PRO Kilépve az utcára Marit rögtön megcsapta a fagyos szél.
 c. PRO Kilépve az utcára Marinak rögtön eszébe jutott, hogy mit felejtett otthon.

Az állapothatározói szerepű határozói igevevek esetében [É. Kiss et al. \(2003\)](#) és [É. Kiss \(2004\)](#) szerint ezzel szemben a határozói igenév alanyának referenciája nem feltétlenül a ragozott ige alanyáéval azonos (31a. és 31b. példák).

- (31) a. Anya a szobában lefektetve hagyta a kisfiát.
 b. Anya a szobában duzzogva találta a kisfiát.

A 32. példa alapján felmerülhet, hogy az állapothatározói szerepű határozói igeveveket tartalmazó mondatokban a PRO alany referenciájának a feltárása hasonló feladat, mint az 5. fejezetben ismertetett Winograd-sémák esetében. Ebben az esetben PRO referenciájának meghatározásához a világismeretre szükséges támaszkodni – miszerint az orvosi rendelőben a betegek szoktak levetkőzni, hogy megvizsgálhassák őket –, a szintaxis és egyéb grammatikai információk nem elegendők a

7.1. Zérónévmások és testetlen alanyok

mondat megértéséhez. A jelenség érdemes lehet arra, hogy a segítségükkel a Winograd-sémák mintájára vizsgáljuk a nyelvi modellek nyelvértési képességeit.

(32) Az orvos PRO levetkőzve vizsgálja meg a betegeket.

Az állapothatározói szerepű határozói igenevek esetében elképzelhető szemantikai többértelműség is, amikor már a világismeret sem nyújt segítséget a referencia feloldásához. A 33. példában a határozói igenév alanya koreferens lehet akár a ragozott ige alanyával, akár a tárgyával. Ilyenkor a beszélő és a hallgató közös tudása, valamint a kommunikációs helyzet segíthet a referencia feltárásában.

(33) Szemüveget viselve láttalak.

A fentiek alapján elmondható, hogy a KorKor korpuszban a határozói igenevek esetében is lehetne a ragozatlan infinitívuszok mintájára zéró alanyt feltüntetni, azonban a korpusz építésekor ezekre az esetekre még nem gondoltam. A korpuszban összesen 63 határozói igenév van, így ezek esetében akár utólag is be lehet illeszteni a zéró alanyokat. Ez nem csupán további tokenek beszúrását jelentené, a tokeneket a függőségi fába is be kellene illeszteni, a megfelelő morfoszintaktikai jegyekkel felruházni őket, valamint beilleszteni őket a meglévő anaforikus kapcsolatok láncolatába.

A határozói igenevek mellett a melléknévi igenevek is szóba jöhetnek mint PRO alannyal rendelkező összetevők. A melléknévi igenevek PRO alanya azzal a főnévi csoporttal koreferens, amelyet a főnévi igenév módosít. Ezt a kapcsolatot ábrázolja a melléknévi igenevek jelzői pozíciója is, ez a viszony pedig könnyedén kinyerhető a függőségi elemzésből is, ezért nem tartom indokoltnak PRO alanyok bevezetését a korpuszba a melléknévi igenevek esetében.

A KorKor korpuszban tehát a ragozott ige zéró alanyát¹, tárgyát, a birtok zéró birtokosát, valamint az infinitívusz PRO alanyát annotáltuk. A következőkben ismertetett zérónévmás-beszűrők is szintén ezeket a zéró elemeket célozzák. Az elsőként bemutatott szabályalapú zérónévmás-beszűrő éppen azért ugyanezeket, mert ezt az eszközt használtam a korpusz építésekor előannotálásra, a másodikként bemutatott neurális megoldás elkészítéséhez pedig a KorKor korpuszt használtam tanítóanyagként, tehát szintén ugyanezeket a zéró elemeket célozza.

¹A zérónévmás-beszűrő kimenetének ellenőrzésekor az annotátoroknak figyelniük kellett arra, hogy azokat a zéró alanyokat, amelyeket pl. a *fog*, *kell* segédigéhez illesztett be a zérónévmás-beszűrő, kitoröljék.

7.2. Szabályalapú zérónévmás-beszúrás: emZero

7.2. Szabályalapú zérónévmás-beszúrás: emZero

A emZero² a KorKor építéséhez készült szabályalapú zérónévmás-beszúró alapján készült, a program az `emt sv` moduljaként is futtatható.

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
- és ragozott és ragozatlan infinitívusz alanyának.

Bemenete a tokenizált és 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). Az algoritmus az emTag által kibocsátott morfológiai címkéket és az emDep kimeneteként kapott függőségi fákat vizsgálja.

A zérónévmások plusz tokenként – tehát új sorokban – jelennek meg az `xt sv` fájlban. Mivel ezek a plusz tokenek névmások, ezért mindegyik szófajcímkéje PRON. A beillesztett névmás a szófajcímké mellett morfológiai jegyeket is kapnak. Például az emZero által beillesztett alany a finit ige ragozása alapján kap személyt és számot, alanyként pedig természetesen nominatívuszban fog állni.

- (34) a. Elment bevásárolni. → alany: E/3
 b. Elmentem bevásárolni. → alany: E/1
 c. Elmentetek bevásárolni. → alany: T/2

Az emZero tárgyat illeszt be a határozott ragozású igéhez, ha a mondatfában nem volt tárgy. A beillesztett tárgy akkuzatívuszban fog szerepelni. A zéró tárgy mindig egyes számú, hiszen a többes számú tárgy mindig megjelenik testes tárgyként. A tárgy lehet második vagy harmadik személyű is, ez kiderül az ige definitívusát kifejező jegyéből.

- (35) a. Peti megette. → tárgy: E/3
 b. Szeretem. → tárgy: E/3

²<https://github.com/vadno/emzero>

7.2. Szabályalapú zérónévmás-beszúrás: emZero

- c. Szeretlek. → tárgy: E/2

A birtokos száma és személye pedig kiderül a birtok megfelelő jegyéből.

- (36) a. Az apja megérkezett. → birtokos: E/3
 b. Az apád megérkezett. → birtokos: E/2
 c. A szüleik megérkeztek. → birtokos: T/2

A zérónévmások beillesztése után a mondatfában plusz ágak jelennek meg, ami azt jelenti, hogy a függőségi mondatelemzéskor kiosztott ID-k közé be kell illeszteni az új elemek ID-it is. A zéró elemek is saját ID-t kapnak, az `xtsv` fájlba 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.

7.2.1. Kiértékelés

A zérónévmás-beszúró kiértékeléséhez³ a KorKor korpusz egy részét használtam. A kiértékelendő anyag az emZero kimenete. A gold standard adat KorKor korpusz építéséhez használt, kézzel javított kimenet. A vizsgálathoz használt anyag méretét a 7.1 táblázat tartalmazza.

	darabszám
fájl	95
mondat	1 436
token	31 492

7.1. táblázat. Az emZero kiértékeléséhez használt anyag mérete a gold standard adaton számolva. A tokenszámba az írásjeleket is beleszámítottam.

Az összevetéshez a 3.3.1. fejezetben ismertetett emDiff kiértékelő modulját használtam. A bemenete `.xtsv` formátumú szöveg. Az emDiff a címkék kiértékeléséhez a pontosság, fedés és F-mérték (F-measure) mérőszámokat használta. Az eredményeket a 7.2 táblázat tartalmazza. Az **összesített**

³A kiértékelést 2019 október 10-én végeztem, az eredményeket közzétettem a GitHubon: https://github.com/vadno/korkor_kiserletek/tree/master/kiertekeles/emzero_eval.

7.3. Zérónévmás-beszűrés nyelvmodell finomhangolásával

eredmény a teljes anyagra számolt pontosságot, fedést és F-mértéket mutatja, az **átlag** pedig a fájl-onként elért eredmények átlaga.

	átlag	összesített
pontosság	93,08%	92,26%
fedés	85,30%	85,25%
F-mérték	88,16%	88,61%

7.2. táblázat. Az emZero teljesítménye pontosságban, fedésben és F-mértékben kifejezve.

A kapott eredményeket a következő alfejezetben bemutatott neurális megoldás részletezése és kiértékelése után, a 7.4. alfejezetben a két módszert összevetve értelmezem.

7.3. Zérónévmás-beszűrés nyelvmodell finomhangolásával

Ebben a fejezetben egy kísérletet mutatok be, amelyben a huBERT (Nemeskey, 2021) modellt finomhangoltam a zérónévmás-beszűrés feladatára. Mivel eddig nem létezett olyan megoldás, amely neurális módszerrel illeszt be zérónévmásokat magyar szövegekbe, ezért ez a kísérlet pótolja a hiányt. A módszer előnye az, hogy tudásszegény, tehát nem igényel egyéb nyelvi elemzést a bemeneti szövegeken.

A zérónévmások beszűrésa azt jelenti, hogy plusz szavak (névmások) jelennek meg a kimenetben. Ebben a kísérletben a zérónévmás-beillesztés problémáját címkézési feladatként valósítottam meg. A projekt GitHub repozitóriumában⁴ elérhető a kísérlet során elkészített modell, valamint a reprodukálhatóság érdekében ugyanitt közzétettem a megfelelő formátumra alakított korpuszt, valamint a teljes folyamatot az adatok előkészítésével, magával a finomhangolással és a kiértékeléssel együtt egy Google Colab munkafüzet formájában. Elsőként ismertetem a finomhangoláshoz előkészített tanítóanyag előkészítését majd bemutatom a finomhangolás folyamatát, végül pedig kiértékelem az eredményeket.

Amint a 3.3.10. fejezetben már ismertettem, a KorKor korpusz tartalmazza a kézzel beillesztett zérónévmásokat az alábbi esetekben:

⁴https://github.com/vadno/bert_zero_hu

7.3. Zérónévmás-beszúrás nyelvmodell finomhangolásával

- 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 korpuszban ennek megfelelően plusz tokenek jelentek meg ott, ahova beillesztettük a névmásokat. Ebben a kísérletben azonban címkézési feladatnak tekintem a problémát, ezért a KorKor formátuma átalakításokat igényelt. A címkék azt fejezik ki, hogy az adott tokenhez tartozik-e zérónévmás, tehát a függőségi fában szerepel-e a token gyermekcsomópontjai között zérónévmás. A címke a zérónévmások fajtáit (alanyi, tárgyi, birtokos) is ábrázolják. Az átalakítás után az egyes tokenekhez az alábbi címkék párosulnak:

- NOZERO: a tokenhez nem tartozik zérónévmás
- SUBJ: zéro alany tartozik hozzá
- OBJ: zéro tárgy tartozik hozzá
- SUBJOBV: zéro alany és tárgy is tartozik hozzá
- POSS: zéro birtokos tartozik hozzá

A finomhangoláshoz a KorKort anyagát használtam, de némi átalakítást végeztem rajta. A finomhangoláshoz használt korpusz nem tartalmaz semmilyen nyelvi annotációt a zérónévmásokon kívül, csupán a szóalakokat. Ennek két oka van: egyrészt hogy olyan alapanyagon is lehessen használni, ami nem tartalmaz egyéb nyelvi elemzést, másrészt pedig hogy egyszerűbb legyen maga a finomhangolás is. A KorKor 80%-át használtam tanítóanyagként a finomhangoláshoz, 20%-ot félretettem a kiértékeléshez.

7.3.1. A finomhangolás

A finomhangolás folyamatának felállításához egy másik, címkézési feladatra készített és közzétett finomhangolási folyamatból⁵ indultam ki. A kiindulási alapként használt megoldás ugyan a névelemcímkézés feladatát célozza, mégis könnyen átalakítható volt a céljaimra, ezért sok kódrészletet tudtam közvetlenül használni. Az általam felállított munkafolyamat néhány ponton eltért a kiindulási alapként használt projekttől. Először is az én megoldásomban a magyar huBERT modellt és a tokenizálóját használtam, valamint az általam használt korpusz az eltérő formátuma mellett ter-

⁵https://github.com/NielsRogge/Transformers-Tutorials/blob/master/BERT/Custom_Named_Entity_Recognition_with_BERT_only_first_wordpiece.ipynb

7.3. Zérónévmás-beszűrés nyelvmodell finomhangolásával

mésztesen a címkekészletében is különbözött a névelemcímkézéshez használt adattól. A projektet tartalmazó Google Colab munkafüzet a lépéseket alaposan dokumentálva tartalmazza, de röviden itt is ismertetem a lépéseket.

A megfelelő csomagok importálása után a munkafüzetbe feltöltöttem a megfelelő formátumban előkészített korpuszt. Az adat formátumának és tartalmának ellenőrzése után a mintaprojektben használt formátumra alakítottam az adatot. Az adat előkészítése után betöltöttem a tokenizálót és tokenizálom a szöveget. Ezután tanító- és tesztelőhalmazra osztottam az előkészített adatot, majd beállítottam a tanításhoz szükséges hiperparamétereket. A nyelvmodell betöltése után történt maga a finomhangolás, majd az eredmények kiértékelése. A kiértékelés után kézzel megadott mondatokon közelebbről is megvizsgálhatjuk a zérónévmás-beillesztő teljesítményét.

7.3.2. Kiértékelés

A kiértékelés során a finomhangoláskor megadott 1 *epoch* után a *training accuracy* 0,96, a *test accuracy* pedig 0,98 lett. A tokenizálás eredményeként kapott *subword tokenek* visszaállítása után az egyes tokenekhez tartozó első subword tokenhez kapott címkét vettem figyelembe a kiértékelés során. A négy címkére külön számítottam pontosságot, fedést és F-mértéket úgy, hogy az egyes címkék esetében az alábbi találati kategóriákba soroltam a modell által kibocsátott címkéket, összevetve a tesztadatban szereplő gold standard címkékkel:

- valós pozitív: a jóslott címke és a gold standard címke megegyezik
- álpozitív: a jóslott címke eltér a gold standard címkétől (és nem NOZERO)
- álnegatív: a jóslott címke NOZERO, a gold standard nem
- valós negatív: minden egyéb esetben

A kiértékelő metrikák eredménye címkénként, összesítve, és a négy címke átlagát tekintve a 7.3. táblázatban látható. Az összesített eredményeknél a címkénként kapott valós pozitív/negatív és álpozitív/álnegatív találatokat összesítettem, majd ezekre számoltam pontosságot, fedést és F-mértéket, az átlageredmény pedig a négyféle címkére kapott pontosság, fedés és F-mérték átlagát jelenti.

Az eredményeken látható, hogy a modell az OBJ címkék esetében teljesített a leggyengébben, azonban szem előtt kell tartani, hogy a tesztadat 260 mondatában összesen 9 darab OBJ címke szerepelt. A leggyakoribb címkéből (SUBJ) 188 szerepelt a tesztadatban, amelyeket nagy pontossággal tudott azonosítani a modell, igaz, a fedése nem volt tökéletes. A négyféle címkére kapott pontosság,

7.4. Diszkusszió

	előfordulás	pontosság	fedés	F-mérték
SUBJ	188	1,00	0,96	0,98
OBJ	9	0,33	1,00	0,50
SUBJOBJ	11	0,73	1,00	0,84
POSS	81	1,00	0,94	0,97
összesített		0,97	0,96	0,96
átlag		0,68	0,72	0,69

7.3. táblázat. A zérónévmás-beszűrő teljesítménye pontosságban, fedésben és F-mértékben kifejezve. Az előfordulás az adott címke darabszámát mutatja a gold standard tesztadatban.

fedés és F-mérték átlaga épp az OBJ címkéknél mért gyenge eredményeknek köszönhetően nem túl magas.

7.4. Diszkusszió

Az emZero és a neurális zérónévmás-beszűrő teljesítményét a kiértékeléskor használt eltérő tesztadat miatt nem lehet közvetlenül összehasonlítani. A két megoldást az általuk célzott feladatokat szem előtt tartva érdemes összehasonlítani. Míg a neurális megoldás feladata az volt, hogy megjelölje azokat a szavakat, amelyekhez testetlen névmás tartozik, addig a szabályalapú megoldás emellett az elemzési fába is beilleszti a megfelelő morfoszintaktikai jegyekkel ellátott zérónévmásokat. A két részfeladat jól kiegészítheti egymást, így érdemes lehet kombinálva alkalmazni a két különböző megközelítésű megoldást. Az emZero alacsony fedését orvosolni lehetne azzal, hogy a testetlen névmással rendelkező elemek megjelölését a neurális megoldással végezzük el, majd a zérónévmás elemzési fába illesztést és a morfoszintaktikai jegyek kiosztását szabályalapon oldjuk meg.

Az emZero esetében fontos szem előtt tartani, hogy a szabályalapú megoldás által kibocsátott kimenet minőségét befolyásolja a többi elemzési réteg minősége is, a morfoszintaktikai vagy a szintaktikai elemzés hibái rontják a rájuk támaszkodó szabályalapú zérónévmás-beszűrő teljesítményét is. Noha ez a hatás a tudásszegény neurális megoldás esetében nem jelentkezik, ha a két megoldást kombinálva szeretnénk használni, szintén számolni kell vele. Ezek alapján akkor érdemes az emZero zérónévmás-beszűrőt használni – akár önmagában, akár a neurális megoldással kombinálva –, ha jó minőségű morfoszintaktikai és függőségi elemzéssel ellátott alapanyagunk van.

8. fejezet

Szabályalapú anaforafeloldás magyar szövegeken

Az anaforafeloldás és a koreferenciafeloldás többféleképpen megközelíthető problémák. Ebben a fejezetben ismertetek néhány általam készített megoldást, amelyek szabályalapú módszerekkel oldják meg az anaforafeloldás feladatát.

8.1. Szabályalapú anaforafeloldó: DepEll

Az első kísérletemet a szabályalapú anaforafeloldás területén az ANAGRAMMA nyelvi elemző elméleti kereteihez igazodva végeztem. Ennek a kísérletnek az eredménye a DepEll névre keresztelt szabályalapú anaforafeloldó algoritmus, amely előzménye a később a KorKor korpusz építéséhez használt anaforafeloldó szkriptnek. A következőkben az ANAGRAMMA elemzőt és a DepEll algoritmust (Vadász, 2017) alapján ismertetem.

8.1.1. Az ANAGRAMMA működési alapelvei

Az ANAGRAMMA elemző (Prószéky et al., 2016) egy pszicholingvisztikailag motivált (Indig et al., 2016), performanciaalapú szövegelemző, amely az emberi szövegfeldolgozás mintájára balról jobbra és szavanként elemzi a bemeneti szöveget. Az elemzőrendszer működési alapelvei a *kereslet-kínálat*

8.1. Szabályalapú anaforafeloldó: DepEll

elvű keretrendszer (Prószéky és Indig, 2015), a *tározó* és az *ablak*. Az elemző kimenete egy függőségi éleket tartalmazó gráf.

Az elemző architektúrája párhuzamos, tehát az éppen elemzendő szót a párhuzamosan jelen lévő szálak (morfológiai elemző, különböző grammatikai jelenségeket azonosító szálak, korpuszgyakorlati szálak, anaforafeloldó szálak, fókuszazonosító szál stb.) egyszerre elemzik. A kereslet-kínálat elvű keretrendszerben az elemzés során az egyes tokenek kínálatként jelenhetnek meg – és így más elemek keresleteit kielégíthetik (pl. a melléknévek a főnevek melléknévkeresőit vagy az esetragos főnevek az ige vonzatkeresőit). A keresleteket keresőeljárások valósítják meg, amelyek megszorításokat is tartalmazhatnak (a keresés irányára, a keresett elem bizonyos jegyeire stb.).

A tározó a balról jobbra történő elemzés során az aktuálisan elemzett tokent megelőző tokeneket, azok keresőit, valamint a kész részszerkezeteket tartalmazza. A tározó tartalma lépésről lépésre változik az elemzés során, a mondat végére a tározóban nem maradhat kínálat vagy olyan kereső, amely kötelező elemet keres.

Pszicholingvisztikailag motivált elemzőként az ANAGRAMMA az emberi szövegfeldolgozást kívánja modellálni. Ehhez Frazier és Fodor (1978) Sausage Machine kétfázisú mondatfeldolgozó modelljét vettük alapul. A Sausage Machine első fázisában a szöveges bemenet szócsoportjaiból létrejönnek a frázisok, amelyek a második fázisban kapják meg a szerepüket a mondatban. A Sausage Machine első fázisa – az angol nyelvre kidolgozva – egy hat token méretű előretekintő ablakban dolgozik. A Sausage Machine elvét az ANAGRAMMA elemzőre átültetve a magyar nyelv agglutináló jellege miatt az előretekintő elemzési ablakot három token méretűnek határoztuk meg. Az előretekintő elemzési ablak azt jelenti, hogy a balról jobbra történő elemzés során egy token elemzésekor az elemzést a tározó tartalmán kívül az adott tokent követő két elem is segíti. Ennek jelentős szerepe van az esetegyértelműsítésben (Vadász és Indig, 2018) és a vonzatkeret-egyértelműsítésben (Indig és Vadász, 2016; Vadász et al., 2017).

8.1.2. Névszói frázisok az ANAGRAMMA elemzőben

Az ANAGRAMMA elemzőrendszerben a névszói frázis (NP) kínálatként viselkedő elem. Az NP-t az ige vonzatkeresője¹ fogja keresni a mondatban, ami az NP fején lévő esetrag alapján történik. A balról jobbra történő kétfázisú elemzésnek általában az első fázisában történik az NP-k előkészítése a

¹Vonzatkeresőnek azokat a keresőket nevezem, amelyek a vonzatkerettel rendelkező elemek összes bővítményét keresik, ideértve a kötelező és opcionális vonzatokat, valamint a szabad határozókat is.

8.1. Szabályalapú anaforafeloldó: DepEll

kínálati szerepre. Az első fázisban történik a szófai egyértelműsítés és a morfológiai elemzés (amelyek nem képezik jelen dolgozat tárgyát). Ezután kétféle problémával kell megküzdeni: (1) a testetlen esetragok kezelése és (2) a többértelmű esetragok egyértelműsítése.

Az NP-t mindig esetrag zárja, tehát a balról jobbra történő elemzés során az elemző arra törekszik, hogy az esetrag megtalálásával lezárja az aktuális NP-t, így előkészítve azt kínáltként a vonzatkerecsők számára. Az alanyesetű névszók esetében azonban egy testetlen esetrag zárja az NP-t. Amikor az elemző egy testes esetrag nélküli elemmel találkozik, nem lehet biztos benne, hogy az alanyeset testetlen esetragjáról van-e szó, vagy pedig az elem nem zárja le a frázist (pl. a jelöletlen birtokos szerkezet vagy a névutós névszók esetében).

Az elemző a kétfázisú mondatelemzés első fázisában az előretekintő elemzési ablak segítségével deríti ki, hogy a testes esetrag nélküli elem NP-záró nominatívusz esetű elem-e. Ehhez az adott elemtől közvetlenül jobbra elhelyezkedő más elemekre támaszkodik. Például ha mondathatár vagy finit ige (vagy bármilyen, tipikusan NP-n kívüli elem) követi, akkor biztos, hogy nominatívusz esettel lezárt NP-vel találkoztunk. Ha azonban a kérdéses esetű elem elemzési ablakában egy birtokos ragozást magán viselő elem látszik – amely birtokos ragozás a kérdéses elemmel egyező birtokost sejtet –, akkor a kérdéses elem lehet egy jelöletlen birtokost tartalmazó birtokos szerkezet birtokosa.²

Külön problémát jelentenek a névutót tartalmazó szerkezetek is. A névutók a testes esetragokhoz hasonlóan kezelendők, hiszen az NP a névutóval való összekapcsolódás után válik kínálattá. A névutót megelőző főnévi fej bizonyos névutók esetében nominatívuszi esetben áll, míg más névutók esetében valamilyen testes esetragot visel magán. A névutós névszók esetében is az előretekintő elemzési ablak nyújt megoldást, hiszen ebben az esetben is a testetlen esetrag tisztázására van szükség. A névutók megfigyelt viselkedése alapján a kérdéses tokenhez nagyon közel, általában közvetlenül utána helyezkednek el, így nem okoz problémát az elemzési ablakban történő kezelésük. Az elemzés további lépéseiben – miután az elemzési ablakban tisztázódott a testes esetrag nélküli elem aktuális szerepe – a névutó a névszó különleges, külön tokenben álló esetragjaként fog funkcionálni: a névutó alapján lesz a névszói frázisból kínálat, hogy az ige vonzati igényét kielégíthesse. A morfológiai elemző kimenete alapján a névszók címkéi a 8.1. táblázatban látható módon épülnek fel.

A tokenek morfológiai címkéjének jegyei különböző eljárásokat indítanak az elemzés során. A tokenek főkategóriája – a kettőspont előtti jegy – alapján viselkedik kínáltként az adott elem. A további jegyei alapján elégítheti ki a kereső megszorításait (pl. a számra és a személyre). Ezenkívül

²Az elemzési ablak működésének részleteit az eset-egyértelműsítés megoldásában, valamint az azokat megalapozó korpuszméréseket Vadász et al. (2017) részletezi.

8.1. Szabályalapú anaforafeloldó: DepEll

főnév	⇒	CAS/Nom: tő+N(+PropN)+Sg/Pl(+PersSg/Pl1-3)
melléknév	⇒	CAS/Nom: tő+NPMod+Adj+Sg/Pl(+PersSg/Pl1-3)
számnév	⇒	CAS/Nom: tő+NPMod+Num+Sg/Pl(+PersSg/Pl1-3)
mn. igenév		
folyamatos	⇒	CAS/Nom: tő+NPMod+Part+Pres+Sg/Pl(+PersSg/Pl1-3)
befejezett	⇒	CAS/Nom: tő+NPMod+Part+Past+Sg/Pl(+PersSg/Pl1-3)
beálló	⇒	CAS/Nom: tő+NPMod+Part+Fut+Sg/Pl(+PersSg/Pl1-3)

8.1. táblázat. A névszók morfológiai elemzését tartalmazó címkék.

bizonyos jegyek alapján a tokennek saját keresői is elindulhatnak, például az olyan, saját vonzatterrel rendelkező elemek esetében, mint a melléknévi igenevek.

A címkéket minden esetben az esetrag jelölése indítja. Az esetrag jelölése a testes ragok esetében az esetrag nevének hárombetűs rövidítése (pl. Acc, Dat, Ela, Sub stb.), a testetlen esetragok esetében Nom. A testes esetragok a hierarchikus jegyszerkezetben a CAS főkategória alá esnek (így minden olyan folyamatot el tudnak indítani, amelyet az esetraggal rendelkező elemek szoktak, pl. a névelőkeresés). A Nom jegy nem tartozik a CAS főkategória alá, hiszen – amint fentebb már kifejtettem – előbb még tisztázni kell, hogy egy nominatívusz esetű NP feje-e, vagy nem. Csak az esete tisztázása után tudja az esetraggal rendelkező elemek által indítandó folyamatokat elindítani, legyen szó akár tisztázott nominatívuszi esetről, akár névutós névszóról.

A címke kettőspont utáni részében a tő jelölése után a lexikális elem aktuális jegyei sorakoznak. A hagyományos szófajokat kiegészítve hierarchikus jegyszerkezetek fejezik ki az adott szófajokra jellemző tulajdonságokat. A főnevet módosító összes szófaj (Adj, Num, Part) rendelkezik a közös NPMod címkével, ami azt jelenti, hogy ők mindannyian a főnevet balról módosító elemek. A főnevek címkéje között megjelenhet a PropN jegy a tulajdonnevek esetében, amelynek a névelőkeresés és a definitiség beállítása során van jelentős szerepe. Az összes névszó rendelkezik számot kifejező jeggyel (Sg vagy Pl), valamint mindegyikhez járulhat birtokos ragozást kifejező jegy is (Pers), amely a keresett birtokos számára (Sg vagy Pl) és személyére (1, 2 vagy 3) tesz megszorítást.

A koreferenciafeloldáskor fontos információ az NP címkéjében található jegyek összessége, de ezeket a morfológiai jegyeket néhány további jeggyel is ki kell egészíteni. A kétfázisú mondatelemzés első fázisában lezárt – tisztázott esetű – NP-k egy, az összes szükséges információt tartalmazó jegyszerkezettel jelennek meg kínálatként a második fázisban működő anaforafeloldó szál keresői számára. A koreferenciafeloldó számára rendelkezésre álló kínálat az alábbi jegyszerkezetekkel rendelkező ún. Szereplő.

8.1. Szabályalapú anaforafeloldó: DepEll

Szereplő	Grammatikai jegyek	CAS = [Nom/Dat/Acc...] szófaj = [N/NPMod(Adj/Num/Part)] szám = [Sg/Pl] személy = [1/2/3] definitség = [Def/Indef] birtokos ragozás = [Pers+Sg/Pl+1/2/3] tul.név = [+/-PropN]
	Szemantikai jegyek = [+/-animate]	
	Aktuális jegyek	funkció = [Subj/Obj/Arg] mondat sorszama = [Sent1...] NP sorszama = [NP1.../null]
	Teljes NP = lemma	

8.2. táblázat. A Szereplők jegyszerkezete, amellyel a koreferenciakeresésben résztvevő elemek rendelkeznek.

A Grammatikai jegyek mellett (ami a morfológiai elemzéstől kinyert címke tartalma alapján töltődik ki) megjelennek Szemantikai jegyek is. Az elemző jelenlegi állapotában csak a [+/-élő] jegy tartozik a szükséges szemantikai jegyek közé. A [+/-élő] jegy a magyar nyelv esetében az ember kategóriára vonatkozik, nyelvi szempontból az állatok [-élő] jegyűek (Frawley, 1992). Az elemzés során a vonzatkeret lehívásakor szükség lehet arra az információra is, hogy az ige egyes vonzataira vonatkozik-e [+/-élő] megszorítás. Emellett a szótár a főnevek [+/-élő] szemantikai jegyeit is kell hogy tartalmazza, hiszen jelentősége lehet a koreferenciafeloldáskor. A főnevek [+/-élő] jegyének megállapításánál a MetaMorpho rendszer (Tihanyi és Merényi, 2006) lexikonjára támaszkodunk.

A grammatikai és a szemantikai jegyek mellett az Aktuális jegyek is a koreferenciafeloldáshoz szükséges információkat szolgáltatnak, közöttük az NP mondatbeli funkcióját (alany: Subj, tárgy: Obj vagy egyéb vonzat: Arg) és a definitségét (Def vagy Indef) kifejező jegyeket. A koreferenciafeloldás során az NP több mondatban is kaphat funkciót, ennek megfelelően az aktuális funkciójához jegyezzük a mondat sorszámát is, valamint azt is, hogy a mondat hányadik NP-jeként töltötte be az adott funkciót (ennek értéke lehet Null is).

8.1. Szabályalapú anaforafeloldó: DepEll

névmási anafora	<i>A gyerek megpusztilta a nénit. Ő meg elmosolyodott.</i>
zéró anafora	<i>A gyerek megölelte a tanárt. Leült és elmosolyodott.</i>
ismétlés	<i>A gyerek meglátta a kutyát. A kutya ugatni kezdett.</i>
alá-, fölérendelés	<i>Laci vett egy pulit. A kutya sokat ugat.</i>
szinonimia	<i>Feri meglátta a rendőrt. A zsarú hirtelen megfordult.</i>
epitheton	<i>Ede elhagyta Marit. A disznó még el sem köszönt.</i>
valószínű rész	<i>A rendőr belépett a szobába. Az ablak nyitva volt.</i>
szükségszerű rész	<i>Tegnap lerobbant a kocsim. A fék romlott el.</i>
esetkeret	<i>A rabló lelőtte a pénztárost. A pisztoly a földön hevert.</i>

8.3. táblázat. Visszaulási típusok Pléh (1998) példáival.

8.1.3. A Pléh-Radics algoritmus

Pléh (1998) alapján a visszaulási kifejezések egyik végpontját az eredetileg deiktikus, indexikus elemek képviselik (*az, ő, ez, oda* stb.), míg a másik végponton a szemantikailag tartalmas, helyettesítő szerepű elemek helyezkednek el.

A proformás névmások működésének alapelve az, hogy használatukkal a szöveg lineáris sorrendjében közelebb kerülnek egymáshoz az azonos referenciájú elemek. Ebben az alfejezetben szabályalapú, algoritmikus megoldást ismertetek, amely az input nyelvtani elemzővel történő feldolgozása révén rendelkezésre álló szintaktikai információkat használ. A Pléh-Radics algoritmus a proformás anaforák használata mögött húzódo szabályrendszert implementálható algoritmusként mutatja be, amely elsősorban a mondatban szereplő névmások és NP-k felszíni nyelvtani tulajdonságaival operál. A 8.3. táblázatban felsorolt visszaulási-típusok közül jelen dolgozatban a zéró és névmási anaforával foglalkozom részletesebben. Ezeket összefoglaló néven proformás anaforának nevezzük. A többi visszaulási-típus kezeléséhez szükség van olyan információkra, amelyhez nem elegendő a morfológiai elemző kimenete.

A (zéró vagy testes) névmások használata mögött egy feltárható szabályrendszer húzódik, amely lehetővé teszi az elemek vonatkozásának visszaállítását. Megkísérelték ennek a szabályrendszernek a feltárását, amelyre a továbbiakban Pléh–Radics algoritmusként fogok hivatkozni. Az algoritmus a mondat alanyának, tárgyának vagy egyéb argumentumának előző mondatbeli antecedensét keresi.

Az algoritmus alkalmazása azokban az esetekben érdekes, amelyekben az anaforikus vagy zéró névmást megelőző szövegrészben több főnévi csoport szerepel, hiszen a szabályrendszer azt határozza meg, hogy a lehetséges jelöltek közül melyik főnévi csoport a névmás antecedense.

8.1. Szabályalapú anaforafeloldó: DepEll

az 1. mondat NP-je		a 2. mondatban	
szerep	egyéb jegy	szerep	megjelenés
alany		alany	∅
tárgy egyéb arg.			'az'
alany	[-élő]	tárgy	∅
tárgy	[+élő]		∅ vagy szem. nm.
egyéb arg.	többes		szem. nm
alany		egyéb arg.	szem. nm.
tárgy			
egyéb arg.			

8.4. táblázat. A Pléh–Radics-algoritmus működésének összefoglaló táblázata.

A 8.4. táblázat a Pléh–Radics-algoritmus működését mutatja be. A második mondat alanya zéró névmás akkor, ha koreferens az előző mondat alanyával. Ezt a jelenséget alanytörlésnek nevezik, és több predikátum esetén is fennáll. Ha az előző mondat tárgyával vagy egyéb argumentumával koreferens, akkor az alanyváltást az 'az' távolra mutató névmás jelzi. Az alany előző mondatbeli antecedensének megkeresése nem függ az előző mondatbeli főnévi csoport szemantikai jegyeitől.

A második mondat tárgyának antecedenskeresése ezzel szemben függ a főnévi csoport [+/-élő] szemantikai jegyétől. Ha a második mondat tárgyát zéró névmás fejezi ki, akkor az előző mondat [-élő] jegyű főnévi csoportjával lehet koreferens. Ha személyes névmás fejezi ki, akkor az előző mondat egy többes számú főnévi csoportjával lehet koreferens. Ha azonban a második mondat tárgya az előző mondat tárgyával koreferens, akkor az zéró és személyes névmásként is realizálódhat. Ez azt jelenti, hogy a második mondat tárgya előző mondatbeli antecedensének meghatározása a többértelműség miatt problémás, ugyanakkor a tárgyi szerepű személyes névmás antecedenskeresését segíti a számbeli és személybeli egyezés az előző mondat valamely nem alanyi argumentumával.

A második mondat egyéb argumentumának koreferenciáját az előző mondat bármilyen szerepű főnévi csoportjával személyes névmás fejezi ki, így ezekben az esetekben a számítógép a számbeli és személybeli egyezésekre támaszkodhat.

A fentieknek megfelelően a Pléh–Radics-algoritmus a koreferenciafeloldás folyamatát a hallgató szempontjából egy többlépcsős döntési algoritmussal írja le (Pléh és Radics, 1976, 275. oldal):

8.1. Szabályalapú anaforafeloldó: DepEll

1. Keresd meg a mondat alanyi funkcióban levő főnévi csoportját; ha ez zéró, akkor az alany megfelel az előző mondat mélyszerkezetei alanyának; ha pedig 'az', akkor az előző mondat valamelyik nem alanyi főnévi csoportjának felel meg.
2. Ha az állítmánynak más vonzata is névmási vagy zéró formában van, akkor annak az előző mondat valamelyik főnévi csoportja felel meg, mégpedig úgy, hogy ha 1. fennáll, akkor a másik.
3. Ha a második mondat alanya nem azonos az előző mondat alanyával, akkor a másodikban levő személyes névmást inkább az első mondat alanyával értelmezzük koreferensnek.

Ezek mellé a perceptuális-értelmezési szabályok mellé járul a mélyszerkezeti alany megkeresésének problémája, ezt azonban nem kezeli a DepEll algoritmus.

8.1.4. Koreferenciafeloldás az ANAGRAMMA elemzőben

A fejezet azt mutatja be, hogy hogyan lehet a Pléh–Radics-algoritmust az ANAGRAMMA balról jobbra és szavanként elemző keretrendszerében alkalmazni. A kétfázisú mondatelemzés első fázisában az egyértelműsített esetű NP-k a 8.1.2. alfejezetben ismertetett módon Szereplővé válnak, tehát kínálatként készen állnak a második fázisban zajló anaforafeloldó szál keresői számára.

A koreferenciafeloldás a vonzatkereséssel összefonódva, azzal párhuzamosan történik. Az igei elem vonzatkeret-egyértelműsítése (Vadász et al., 2017) után kezdődik a vonzatok keresése, megfelelő megszorításokkal a keresett elemekre. A következőkben az alany és a többi vonzat keresésével párhuzamosan történő koreferenciafeloldást ismertetem. A példákban a finit ige vonzatait keresem, az infinitívusz és a többi vonzatkerettel rendelkező elem vonzatkeresését ebben az alfejezetben nem tárgyalom.

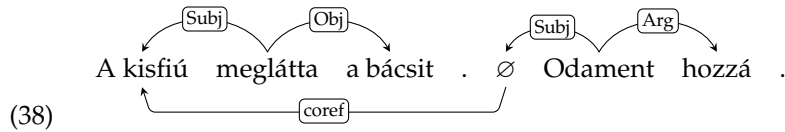
Az alfejezetek példái a vizsgált szempontok alapján létrejövő függőségi éleket szemléltetik. A példamondatok fölötti kerek élek a mondatok főigéje és vonzatai között létrejövő vonzatéleket reprezentálják, a példamondatok alatti szögletes élek a mondatok szereplői között létrejövő koreferenciaéleket.

8.1.5. Az alany keresése

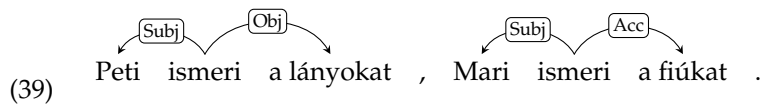
Az elemző alanykereséskor a tisztázott nominatívusz esetű elemeket keresi. Az alanykeresés legelső lépéseként a finit ige ragozásából kiszámolja a keresett alany számát és személyét, és létrehoz egy

8.1. Szabályalapú anaforafeloldó: DepEll

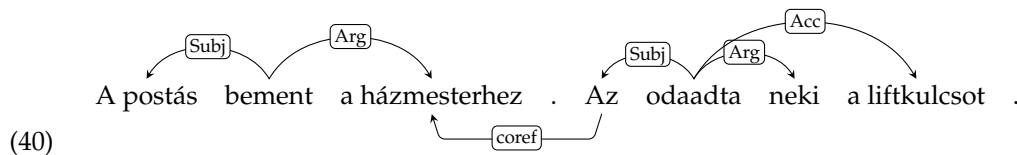
Az ismétlődő alany törlésének szabálya a Pléh-Radics algoritmus alapján: ha volt előző mondat, és a finit ige ragozása az előző mondat finit igéjének ragozásához képest nem változott, akkor a zérus csomópont koreferens lesz az előző finit ige alanyával. Ezt az esetet a 38. példa illusztrálja.



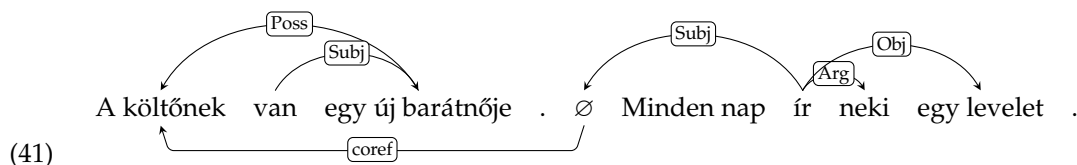
Ha az aktuális mondatban az elemzés során érkezik egy tisztázott nominatívusz esetű NP, amely szám és személy jegyeivel megfelel az ige által indított alanykereső megszorításainak, akkor az Szereplőként átveszi az alany szerepét a zérus csomóponttól (és Subj jegyet kap). A 39. példán látható, hogy a második mondata nem koreferens az előző mondat egyik Szereplőjével sem.



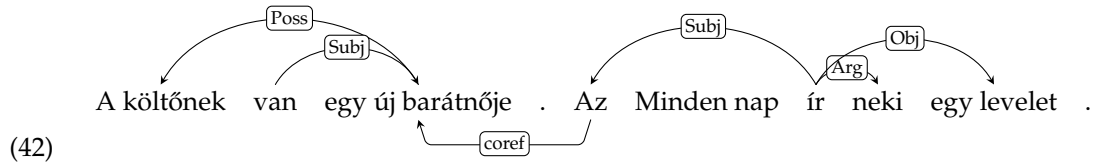
Egyébként ha a mondatban szerepelt egy egyértelműsített *az* távolra mutató névmás, akkor átveszi a zérus csomóponttól az alany szerepét (és Subj jegyet kap), valamint koreferenciaél jön létre közte és az előző mondat egy Arg jegyű elemével. Ezt az esetet mutatja be a 40. példa.



Mivel az alanykeresés az elemző jelenlegi állapotában csak a tisztázott nominatívusz esetű elemekre korlátozódik, ezért a datívusz esetű mélyszerkezeti alany problémájával nem képes megküzdeni. Pléh és Radics (1976) megállapította, hogy „a második mondat alanyi főnévi csoportja zéró formában realizálódik, ha az első mondat mélyszerkezeti alanyával koreferens, s mutató névmás (az) formában, ha az előző mondatbeli antecedense a mondatnak ott nem mélyszerkezeti alanya volt.” (Pléh és Radics, 1976, 266). Az előbbi esetre lásd a 41. példát, az utóbbira a 42. példát, amelyek Pléh és Radics (1976) példái. A 41. példában a második mondat zéró alanya az előző mondat datívusz esetű elemével (a mélyszerkezeti alannal), míg a 42. példa távolra mutató névmással kifejezett alanya az előző mondat nominatívusz esetű bővítményével koreferens.



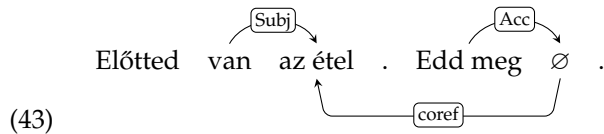
8.1. Szabályalapú anaforafeloldó: DepEll



8.1.6. A tárgy keresése

A kereső a tárgy esetében is az NP-k esetragja alapján keres. A tárgy keresésekor az alanyhoz hasonlóan az elemző a finit ige ragozásából kiszámítja a keresett tárgy számát, személyét és definitségét, valamint létrehoz egy zérus default csomópontot a megfelelő jegyekkel. A zérus csomópont Szereplőként megkapja a jegyszerkezetet és rögtön megkapja a tárgy szerepét (Obj jeggyel).

Ha volt előző mondat, amelyben volt egy Subj jegyű, egyes számú elem, amelynek a jelenlegi mondatban nincs Subj jegye, akkor a tárgy azzal az elemmel koreferens lesz (így az a Szereplő megkapja a második mondat finit igéjének Obj jegyét). A 43. példa egy ilyen esetet mutat be.

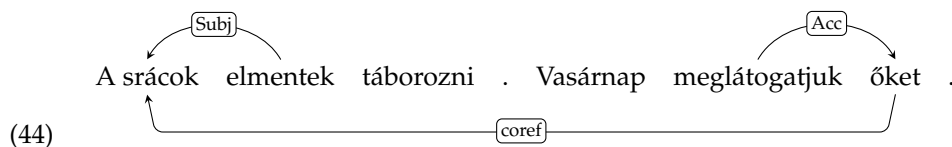


Ebben az esetben a Szereplő jegyszerkezete a 8.2. ábrán látható módon fest.

A 8.2. ábrán látható példa azt illusztrálja, hogy hogyan változik egy Szereplő NP jegyszerkezete mondatról mondatra. Alanyesetű névszói frázisként az első mondatban az alany szerepét töltötte be, majd a második mondatban zéró névmásként a tárgy szerepét.

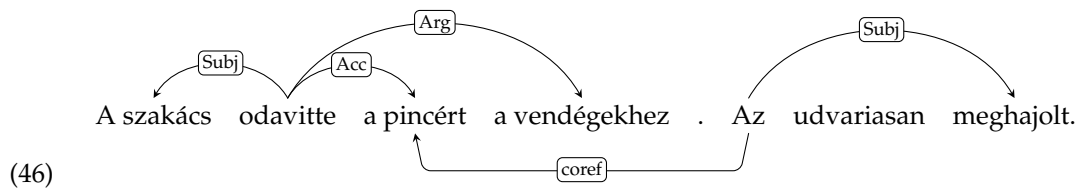
Egyéb esetben az előző mondat egyik nem Subj jegyű elemével lesz koreferens (amely szintén Obj jegyet kap).

Ha az aktuális mondatban van Acc esetű személyes névmás, amely az előző mondat Subj jegyű elemével számban egyezik, és amelynek a jelenlegi mondatban nincs Subj jegye, akkor ezzel az elemmel lesz koreferens a tárgy (és Obj jegyet kap), amint a 44. példán látható.



8.1. Szabályalapú anaforafeloldó: DepEll

A 46. példában látható, hogy az alanyváltás után a második mondat alanya csak az előző mondat tárgyával lehet koreferens, hiszen a másik argumentummal nem egyezik számban.



8.1.8. Többértelmű anafora

A nem alanyi vonzat szerepű proformás anafora antecedensének keresésekor előfordulhat, hogy a megelőző mondatban több lehetséges jelölt is szerepel. Ekkor pusztán a nyelvtanszerű szabályrendszerre támaszkodva nem tudunk dönteni.

A 47. példa második mondatában mindkét proformás anafora esetében több antecedensjelölt is felmerül a megelőző mondatban. Az *az* távolra mutató névmás mint az alanyváltást jelölő elem az előző mondat valamely nem alanyi vonzatával lesz koreferens (*a rendezővel* vagy *az operatőrrel*). A *hozzá* személyes névmás a második mondat egyéb argumentumaként az előző mondat bármely Szereplőjére vissza tud utalni, így koreferens lehet akár a színésszel, akár a rendezővel, akár az operatőrrel. A *rendező* és *az operatőr* a második mondatban természetesen csak egy szerepet kaphat, így ha *a rendezővel* koreferens az *az* távolra mutató névmási alany, akkor *ahozzá* névmás már nem lehet vele koreferens.

(47) A színész szólt a rendezőnek az operatőrrel. Az odajött hozzá.

8.1.9. Diszkusszió

A DepEll algoritmust nem implementáltam, így kiértékelés sem készült hozzá. Az algoritmus azonban alapját képezte a következő fejezetben bemutatott szabályalapú anaforafeloldó programnak, amelyet a KorKor építések előannotálására használtam.

A DepEll algoritmus kidolgozása megkívánta, hogy a Pléh-Radics algoritmusban megjelenő szabályokat úgy alkalmazzam, hogy illeszkedjenek az AnaGramma a bemenetet balról jobbra és szavanként feldolgozó működésébe. Ehhez képest a következők fejezetben ismertetett, a KorKor építéséhez

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

implementált megoldás esetében már nem áll fent az a megszorítás, amely a szöveget balról jobbra és szavanként történő feldolgozása jelent, ez pedig nagyban megkönnyítette a működő szkript elkészítését.

A DepEll algoritmushoz definiált Szereplő adatszerkezet átemelhető volt a működő feloldó programkódjába, azonban míg az AnGramma esetében feltételeztem egy lexikális erőforrást, amely szemantikai információkat is tartalmaz (Tihanyi és Merényi, 2006), a KorKor építésekor elkészített program nem támaszkodik ilyen erőforrásra.

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

A 3. részben ismertetett KorKor korpusz építésekor a névmási anaforákat egy szabályalapú feloldó segítségével előre beillesztettem. Mivel nem állt rendelkezésre olyan anaforafeloldó magyar nyelvre, amely szabadon hozzáférhető lett volna, ezért a 8.1. fejezetben ismertetett DepEll alapján készítettem el azt az eszközt, amelyet a korpuszepítéskor használtam előannotálásra. Az előannotálásnak köszönhetően az annotátoroknak a névmási anaforákat csupán ellenőrizniük és javítaniuk kellett, emellett persze a többi névmástípus és a koreferenciakapcsolatok esetében kézzel kellett beilleszteniük minden kapcsolatot.

8.2.1. Az anaforafeloldó működése

Az előannotáláshoz készített program megkeresi a szövegben a személyes névmásokat (a morfológiai egyértelműsítés alapján), a mondatban szereplő többi szó szófaji, morfológiai és szintaktikai információira támaszkodva egyszerű szabályok alapján dönt. A program a Pléh-Radics algoritmus alapján keresi meg a személyes névmások előzményét és az alábbi szabályokra támaszkodva dönt:

1. ha az ige alanya zéró névmás és az ige ragozása az előző mondat igéjének ragozásához képest nem változott, akkor az előző mondat igéjének alanya az antecedense
2. ha az ige alanya mutatónévmás, akkor az előző mondat nem alanyi argumentuma az antecedense
3. ha az ige tárgya névmás és az előző mondat igéjének valamely argumentumával számban és személyben egyezik, valamint a kérdéses argumentum nem antecedense másnak, akkor az előző mondat kérdéses argumentuma az antecedense

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

4. ha az ige nem alanyi vagy tárgyi egyéb argumentuma névmás és az előző mondat igéjének valamely argumentumával számban és személyben egyezik, valamint a kérdéses argumentum nem antecedense másnak, akkor az előző mondat kérdéses argumentuma az antecedense
5. ha a birtokos névmás és az előző mondat igéjének valamely argumentumával számban és személyben egyezik, valamint a kérdéses argumentum nem antecedense másnak, akkor az előző mondat kérdéses argumentuma az antecedense

Az előzménykereséshez feldolgozza a mondatokban szereplő tokenekre vonatkozó morfológiai és szintaktikai információkat. Az algoritmus működése röviden a következő:

A szöveg feldolgozása során minden tokenhez kinyeri a szükséges morfológiai és szintaktikai információkat és egy megfelelő adatszerkezetbe menti el. A függőségi fák feldolgozásakor minden tokenhez hozzárendeli az összes rámutató függőségi viszonyt – illetve azokat az elemeket, amelyekből kiinduló függőségi viszony mutat az adott tokenre. Ezek közül a függőségi kapcsolatok közül a feladat szempontjából a vonzatok (SUBJ, OBJ, OBL, DAT) és a birtokosok (POSS) a fontosak. Miután előáll a megfelelő adatszerkezet, a fent ismertetett szabályok alapján felállított elágazásrendszer állítja elő a személyes névmások előzményét jelölő kapcsolatokat. A kérdéses névmás előzményének keresésekor mindig megvizsgálja az összes előzményt, megnézi, hogy teljesül-e a számban való egyezés, megnézi, hogy melyik lehetséges jelölt milyen szintaktikai szerepben áll, meg persze azt is, hogy a lehetséges előzményjelölt nincs-e már lefoglalva előzményként egy másik névmás számára.

Az elágazásrendszer az alanyi szerepű személyes névmások esetében például a következő feltételek mentén dönt. Ha a személyes névmás zéró, akkor előzménye az előző mondat alanya. Ha a személyes névmás lemmája az 'az', akkor előzménye az előző mondat nem alanyi vonzata (OBJ, OBL, DAT). A feloldó megtalálható a KorKor elkészítésekor használt szkriptek között a GitHubon³.

8.2.2. Kiértékelés

Az anaforafeloldó kiértékeléséhez a KorKor korpusz építésekor létrejött, kézzel ellenőrzött adatot használtam. Amint a 3.3.12. fejezetben ismertettem, a KorKor építésekor a szabályalapú anaforafeloldó kimenetét annotátorok ellenőrizték és javították. Az annotátorok azonban ebben a lépésben nemcsak az anaforikus kapcsolatokat ellenőrizték és javították, hanem beillesztették az automatikus anaforafeloldó által nem kezelt anaforikus kapcsolatokat és a koreferenciakapcsolatokat is. Tovább nehezíti a kiértékelést, hogy az automatikus anaforafeloldás kimenete és a végleges korpuszfájlok kö-

³https://github.com/vadno/korkor_pilot/blob/master/scripts/anafora.py

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

zött tokenizálási különbségek is lehetnek. Ezek miatt az anaforafeloldó teljesítményének kiértékeléséhez nem elegendő egyszerűen összevetni az anaforafeloldó kimenetét az annotátorok által átnézett fájlokkal.

A kiértékelés első lépéseként a kiértékelendő és a gold standard fájlok tokenizálását egységesítettem. Ennek eléréséhez az emDiff (3.3.1) segítségével előállítottam a kiértékelendő–gold standard fájlok deltáit, aminek segítségével ki tudtam zárni azokat a sorokat, amelyek vagy csak az egyik, vagy csak a másik fájlban vannak jelen. Az összevetéshez így 26 171 token állt a rendelkezésemre.

A gold standard fájlokból – tehát KorKor végleges korpuszfájlaiból – azokat az annotációkat vettem gold standard címkének, amelyek mellett a személyes névmási (p_{rs}) címke jelent meg az anaforikus kapcsolat típusaként. A többi anaforikus- vagy koreferenciakapcsolat nem játszott szerepet a kiértékeléskor, hiszen ezeknek a kapcsolattípusoknak a jelölése nem is volt feladata az automatikus anaforafeloldónak. A gold standard anyagban összesen 1 196 tokent jelöltek az annotátorok antecedensttel rendelkező személyes névmásnak, ebből csupán 82 volt testes névmás.

Az anaforafeloldó teljesítményét igyekeztem több szempontból is kiértékelni. Az első kiértékelésben az eredeti, linkalapú annotációt vettem figyelembe. Itt azt vizsgáltam, hogy az anaforafeloldó megtalálta-e a visszautaló elemeket, valamint a megfelelő antecedenst választotta-e ki a visszautaló elemek számára. Az első feladat esetében – ahol a visszautaló elemek megjelölését vizsgáltam – pontosságot, fedést és F-mértéket számoltam, tehát címkézési feladatként értékeltem a visszautaló elemek megjelölését tekintet nélkül arra, hogy helyesen választotta-e ki az antecedenst a visszautaló elemek mellé. Az anaforafeloldó kimenetét összevetve a gold standard adattal az alábbi kategóriákba osztályoztam az egyes tokeneket:

- valós pozitív: a gold standard adatban és az anaforafeloldó kimenetében is van jelölve antecedenst
- álpozitív: az anaforafeloldó kimenetében van jelölve antecedenst, de a gold standardban nincs
- álnegatív: az anaforafeloldó kimenetében nincs jelölve antecedenst, de a gold standardban van
- valós negatív: a gold standard adatban és az anaforafeloldó kimenetében sincs jelölve antecedenst

Az egyes találati kategóriákba eső tokeneket összeszámoltam, majd pontosságot, fedést és F-mértéket számoltam. Az eredményt összevettem egy egyszerű baseline megoldással is, ahol minden tokent megjelöltem, amelynek névmási címkéje van, vagy ami droppolt névmás. Az eredmények a 8.5. táblázatban láthatók.

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

	pontosság	fedés	F-1
anaforafeloldó	0,86	0,33	0,44
baseline	0,79	0,86	0,94

8.5. táblázat. Az anaforafeloldó teljesítménye pontosságban, fedésben és F-mértékben kifejezve.

Az eredmények azt mutatják, hogy az anaforafeloldónak a fedés és az F-mérték esetében nem sikerült jobban teljesítenie a baseline megoldásnál az előzménnyel rendelkező elemek kijelölésében. Az alacsony fedés oka az lehet, hogy vannak olyan személyes névmási anaforák, amelyek kezelését eleve nem is célozta az anaforafeloldó. Egyedül a pontosság volt valamivel magasabb a baseline megoldásnál ebben az alfeladatban.

A kiértékelés következő szakaszában már azt is figyelembe vettem, hogy az anaforafeloldó helyesen választotta-e ki a visszautaló elem antecedenst. Ennél az alfeladatnál csak pontosságot (accuracy) tudtam számolni, tehát azt számoltam össze, hogy hány esetben szerepelt pontosan ugyanaz az annotáció az adott token mellett, mint a gold standard adatban. Találatnak számított tehát, ha az előzménnyel nem rendelkező token mellett nem szerepelt előzmény-index, és az is, ha az előzménnyel rendelkező token mellett ugyanaz az előzmény-index szerepelt, mint a gold standard adatban. Az előzmények teljesítménye pontosságban kifejezve 95%, tehát 100-ból 95 token mellett meggyezett az anaforafeloldó és az annotátorok által megadott annotáció. Az eredmény értékeléséhez azonban fontos adalék, hogy a pontosan eltalált annotációk 99%-a antecedenssel nem rendelkező token volt, csupán 1%-a mellett jelent meg pontosan ugyanaz az antecedens-index.

Az anaforafeloldó összesen 574 token esetében jelölt meg antecedenst. Az annotátoroknak ezek közül 168-at fogadtak el, ebből 165 zérónévmási anafora volt, valamint néhány esetben testes névmás (*őt, ők*). 225 esetben módosítaniuk kellett a token mellett megadott antecedens-indexet (és esetleg a kapcsolattípust is). Ezek közül 207 zérónévmás volt, valamint néhány személyes névmás és a személyragos névutók (*neki, rajta, belőle, hozzá, érte, róluk, rájuk*). 181 token esetében vagy törölték a kapcsolatot, vagy más kapcsolattípust adtak meg (pl. vonatkozó névmási típust). Az előbbi annotálási hibának tekintendő, hiszen minden névmás előzménnyel kell, hogy rendelkezzen, így a jövőben ezeket a hibákat ki kell javítani és újra elvégezni a kiértékelést.

Az anaforafeloldó hibáinak javítása mellett 803 esetben kellett az annotátoroknak kézzel beilleszteniük a névmási anaforikus kapcsolatokat. Ezek közül 742 zérónévmáshoz, a többi esetben személyes névmásokhoz (*mi, őt*), vonatkozó névmásokhoz (*aki, amelynek, ahhoz*), személyragos névutókhöz (*neki, rajta, belőle, hozzá, érte, róluk, rájuk*, stb. Ezek mellett néhány olyan szóhoz is feltüntettek anafori-

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

kus névmási kapcsolatot, amelyek nyilvánvalóan nem névmások (*várvédőknek, tolvajnak, szkúner, sőt, tízéves*). Természetesen ezek is javítandó annotálási hibák.

A második kiértékeléskor az eredetileg linkalapú annotációt átalakítottam entitásalapúra, hogy megvizsgálhassam a szövegben az azonos referenciájú elemek által alkotott klasztereket. Az eredeti annotáció alapján azonos klaszterazonosítót kaptak azok az elemek, amelyek között anaforikus kapcsolat áll fenn. Több klaszterkiértékelő metrikát⁴ is alkalmaztam. A módosított rand index (*adjusted rand index*, ARI) a klaszter-hozzárendelések közötti hasonlóságot méri páronkénti összehasonlítással úgy, hogy jutalmazza a valós pozitív és a valós negatív eseteket. A V-mérték (*V-measure*) feltételes entrópiaanalízis segítségével méri a klaszter-hozzárendelések minőségét. A magasabb pontszám nagyobb hasonlóságot jelent. Két mérőszámból áll össze: homogenitás és teljesség. A homogenitás akkor magasabb, ha a klaszterek csak egy osztály tagjait tartalmazzák, míg a teljesség akkor magasabb, ha egy adott osztály minden tagja ugyanahhoz a klaszterhez van hozzárendelve. A V-mérték a homogenitás és a teljesség átlaga. A homogenitás, a teljesség és a V-mérték a pontossággal, fedéssel és F-mértékkel analóg módon értelmezhetők. Az eredmények a 8.6. táblázatban láthatók.

metrika	eredmény
ARI	0,32
homogenitás	0,18
teljesség	0,51
V-mérték	0,27

8.6. táblázat. Az anaforafeloldó által produkált klaszterek kiértékelése.

8.2.3. Diszkusszió

Felmerülhet a gondolat, hogy a KorKor építésekor használt automatikus anaforafeloldó a gyenge teljesítménye miatt nem segítette az annotátorok munkáját, valamint nem is gyorsította meg azt. A SzegedKoref esetében sem használtak előannotációt, a humán annotátorok feladata volt az összes anaforikus- és koreferenciakapcsolat beillesztése kézzel. Ugyanakkor azt is szem előtt kell tartani, hogy elsősorban az alacsony pontosságú előelemzés az, ami megnehezítheti az annotátor dolgát, a nagy pontosság és alacsony fedés kevésbé problémás. A korábbi korpuszépítési tapasztalatok azt mutatják, hogy az annotátorok gyakrabban hibáznak, ha egy hibás annotációt kell javítaniuk, ugyan-

⁴A kiértékeléshez a Python scikit-learn csomagját használtam.

8.2. A KorKor építésekor használt szabályalapú anaforafeloldó

akkor természetesen gyorsabban tudnak annotálni, ha van előannotáció. Az előannotáció megtervezése és elkészítése során tehát arra kell törekedni, hogy az előannotáló eszköz a lehető legpon-
tosabban működjön, akár a fedés rovására is. Végül pedig, a KorKor korpusz esetében egy olyan
annotátori visszajelzést sem kaptam, hogy zavaróak lennének az előre beillesztett személyes névmá-
si anaforikus kapcsolatok. Összességében tehát elmondható, hogy az ismertetett anaforafeloldó a
kezelt anaforatípusok szűk köre miatt csak előannotálásra alkalmas.

9. fejezet

Koreferenciafeloldás BERT-tel

Ez a fejezet Nyéki Bencével közös kutatásunkat, egy kísérletet mutat be, amelyben neurális megoldást kínálunk a koreferenciafeloldás feladatának megoldására magyar nyelvű szövegekben. A fejezet az ő hozzájárulásával került a disszertációba, a fejezet a közös munka során készült publikáció (Vadász és Nyéki, 2023) alapján készült. A projekthez az alábbi feladatok elvégzésével járultam hozzá: definiáltam a neurális koreferenciafeloldótól elvárt feladatot; egységes, a feladathoz illő formátumra hoztam a SzegedKoref és a KorKor korpuszokat; tanító-, fejlesztő- és tesztalmazra osztottam a korpuszt; leelemeztem a korpuszokat a szófajalapú előszűréshez; valamint kiértékeltem a szófajalapú előszűrőt.

A kísérletünkben összevonva használtunk két koreferenciakorpuszt a magyar BERT modell finomhangolásához, amivel magyar szövegeken lehet koreferenciafeloldást végezni. A cikk ismerteti a kísérletünk lépéseit a korpuszok előkészítésétől és felhasználásától a BERT modell finomhangolásán keresztül az eredmények kiértékeléséig. A koreferenciafeloldót szabadon hozzáférhetővé tettük.

A neurális nyelvmodellek lehetővé teszik, hogy finomhangolás segítségével magasabb szintű nyelvfeldolgozási feladatokat oldjunk meg. A finomhangolás során egy kisebb méretű, speciális annotációval ellátott korpuszt használunk. Mivel a koreferenciafeloldás feladatának megoldását tűztük ki célként, a finomhangoláshoz koreferenciaannotációval ellátott korpuszra volt szükségünk. Abban a szerencsés helyzetben vagyunk, hogy magyarra már rendelkezésre állnak nagy méretű neurális nyelvmodellek és a kitűzött feladathoz alkalmazható koreferenciakorpuszok is.

Neurális megoldás a legjobb tudásunk alapján magyarra eddig nem született a témában, így a mi megoldásunk ezt a hiányt pótolja. Kísérletünk során a huBERT modellt finomhangoltuk a Sze-

9.1. A felhasznált korpuszok

gedKoref és a KorKor korpuszok felhasználásával. Az általunk készített koreferenciafeloldó szabadon hozzáférhető, a GitHub-on¹ elérhető. A fent ismertetett angol nyelvre készült megoldásokkal szemben a kutatásunkban megpróbáltunk egy konceptuálisan egyszerűbb, könnyebben felépíthető megoldásra törekedni a finomhangolt huBERT modell utolsó rétege által kibocsátott kontextuális tokenbeágyazások klaszterezésével.

9.1. A felhasznált korpuszok

A kísérleteink során két korpuszt használtunk, amelyek többek között koreferenciaannotációt is tartalmaznak, a 3.1. alfejezetben ismertetett SzegedKoref korpuszt, valamint a doktori kutatás keretében készült KorKor korpuszt (3. fejezet). Hogy minél nagyobb tanító- és tesztelőanyag álljon rendelkezésünkre, a két korpuszt egységes formátumra alakítva használtuk fel a kísérletünkhöz.

Mindkét korpusz tartalmazza a zérónévmásokat is, így azokat az anaforikus kapcsolatokat is, amelyekben a zérónévmások vesznek részt. A pro-drop nyelvek esetében nagyon fontos, hogy egy koreferenciakorpusz tartalmazza ezeket a zéró elemeket. A KorKor korpusz esetében például az összes névmás háromnegyede zérónévmás. A zérónévmások mellett minden nem testes elem, tehát a zéró létigék és elliptált igék is nagyon fontos szerepet játszanak a különböző információkinyerési feladatokban. Mindezek ellenére a jelenlegi kísérletben nem használtuk a zérónévmásokat. Ennek az az oka, hogy a koreferenciafeloldó program bemenete sem tartalmazza őket – hacsak nem egy előfeldolgozó lépés eredményeként. A jelenlegi munkában tehát csak a szövegekben testesen előforduló elemekkel dolgoztunk. A SzegedKoref elérhető egy olyan verzióban, ami nem tartalmazza ezeket a testetlen elemeket, a KorKor korpuszt pedig automatikus módszerekkel alakítottuk át ilyenné.

A két korpusz formátuma és annotációs sémája több ponton is eltér, ezért mindenképp egységesíteniünk kellett őket. A KorKor esetében a dependenciaelemzés mintájára a koreferenciakapcsolatban résztvevő elem feje mellett szerepel az antecedens vagy a szövegben korábban előforduló koreferens elem index-száma, míg a SzegedKorefben a konstituenselemzéshez hasonlóan vannak összezárójelezve az összetartozó elemek. Az utóbbi a szerencsésebb megoldás, mert így az egymásba ágyazott vagy mellérendelt főnévi csoportok esetében is egyértelműen kiderül, hogy melyik főnévi csoport szerepel a koreferenciakapcsolatban. A SzegedKoref esetében ez azt jelenti, hogy egy token mellett akár több index-szám is szerepelhet. Az egységesítés azonban sajnos csak úgy volt automatikusan megoldható, ha a KorKorban használt módszert alkalmazzuk mindkét korpuszra. Ez azt jelenti, hogy

¹https://github.com/nytud/bert_coref_hu

9.1. A felhasznált korpuszok

a SzegedKoref esetében néhány koreferenciakapcsolatot elveszítettünk. Ha egy token mellett több koreferens elem is meg van jelölve, akkor csak a legelső főnévi csoportozhoz tartozó index-számot tartottuk meg.

Mindkét korpusz rövid szövegekből áll (KorKor: 112 és 652 token között, SzegedKoref: 11 és 1 496 token között), az előkészítés során mégis tovább kellett darabolnunk őket. A huBERT bemenetét ugyanis ún. subword tokenekre, tehát szótöredékekre kell bontani, a modell pedig csak olyan szekvenciákat dolgoz fel, amelyek legfeljebb 512 subwordöt tartalmaznak. A feldarabolás után 600 darab fájl kaptunk.

A két korpusz méretének és a bennük szereplő szövegek műfajának az arányát megtartva tanító, fejlesztő és tesztalmezra bontottuk őket. A 9.1. táblázat mutatja az egyes halmazok méretét.

forrás / halmaz	tanító	fejlesztő	teszt	ÖSSZESEN
SzegedKoref	98 774	12 131	12 521	12 3426
HVG	7 779	772	1 163	9 714
iskolai fogalmazás (8.o.)	73 891	9 353	9 215	92 459
iskolai fogalmazás (10.o.)	17 104	2 006	2 143	21 253
KorKor	22 222	2 837	2 817	27 876
hír	6 970	887	803	8 660
Wikipedia	15 252	1 950	2 014	19 216
összesen	120 996	14 968	15 338	151 302

9.1. táblázat. A két korpuszból elkülönített tanító, fejlesztő és tesztanyag mérete tokenben kifejezve, a központosítási jeleket is beleszámítva.

A 9.2. táblázatban a korpuszokban jelölt anaforikus és koreferenciakapcsolatok száma látható. Azt is szeretnénk volna megjeleníteni, hogy a korpuszokban összesen hány koreferenciacsoport van. Ez nem azonos a szövegekben előforduló entitások számával, hiszen csak azokat az entitásokat számítottuk bele, amiknek legalább két említése volt egy szövegben. Ennek az az oka, hogy a csupán egyetlen említéssel rendelkező entitások nincsenek megjelölve az általunk használt korpuszokban (lásd a 9.2. fejezetben). Az összevont korpusz fájljai átlagosan 11-12 koreferenciacsoportot tartalmaznak.

9.2. Szófajalapú előszűrés

	KorKor		SzegedKoref	
	kapcsolat	csoport	kapcsolat	csoport
tanítóadat	2 521	796	10 290	3 522
fejlesztőadat	364	113	1 294	432
tesztelőadat	406	121	1 211	431

9.2. táblázat. A kapcsolatok és a koreferenciacsoporthoz való kapcsolatok száma a korpuszokban.

9.2. Szófajalapú előszűrés

Az egyes koreferenciakorpuszok között különbségek lehetnek abból a szempontból, hogy hogyan jelölik azokat a referáló elemeket, amelyek nem állnak semmivel koreferenciakapcsolatban. (A nemzetközi szakirodalomban ezekre *singleton*-ként hivatkoznak, itt a továbbiakban *singleton*-nak nevezzük.) Az általunk használt korpuszokban a singletonok nem voltak annotálva, ezért nem állt a rendelkezésünkre tanítóadat ehhez az alfeladathoz.

A nemzetközi szakirodalom a *mention* vagy *markable* kifejezéseket használja az anaforikus- vagy koreferenciakapcsolatokban potenciálisan részt vevő elemek megnevezésére, a továbbiakban *jelölteknek* nevezzük ezeket. A lehetséges jelöltek kiválogatása egy külön előfeldolgozási lépésként értelmezhető a tényleges koreferenciafeloldás előtt. Megoldásunk azonban más irányból közelíti meg ezt a feladatot: a jelöltek kiválogatása helyett megpróbáljuk kizárni azokat a tokeneket, amelyeknek valószínűleg nincs antecedense vagy nem vesz részt koreferenciakapcsolatban. Szerencsére egyszerű szabályok alapján könnyen szűkíthetjük a lehetséges jelöltek körét. Ehhez szófajalapú előszűrést alkalmaztunk.

Mindkét általunk felhasznált korpusz tartalmaz kézzel ellenőrzött minőségű nyelvi annotációt, így a szófajt is – igaz, eltérő annotációs sémákkal és címkekészletekkel. Az egységesítés kedvéért ezért mindkét korpuszt leelemztük az *emt.sv* elemzővel. Ez azt jelenti, hogy a gold standard minőségű kézi annotációt feláldoztuk az egységesítés oltárán, ugyanakkor azt is szem előtt kell tartani, hogy az *emt.sv* nagyon jó minőségű morfológiai elemzést bocsát ki (Simon et al., 2020).

A nyelvi jelenségek ismeretében és a korpuszok segítségével nagy biztonsággal megállapítható, hogy mely szófajokat érdemes kizárni. Ez az egyszerű előszűrés lépés nagymértékben megkönnyíti a koreferenciafeloldó munkáját, még ha feltételezzük is, hogy az előszűrés során használt szófaji címkézéssel esetleges hibái begyűrűzhetnek a következő feldolgozási lépésekre, így rontva a koreferenciafel-

9.3. Finomhangolás

oldó teljesítményét. Az előszűrés során az igéket, igekötőket, csak jelzőként használt mellékneveket, határozószókat, névelőket, névutókat, kötőszavakat, indulat- és mondatszavakat és az írásjeleket zártuk ki a jelöltek köréből.

9.3. Finomhangolás

A kísérletünkben a huBERT által kibocsátott kontextuális tokenbeágyazásokat finomhangoltuk. A célunk az volt, hogy a tokenbeágyazásokat úgy módosítsuk, hogy megragadják a koreferenciakapcsolatokat, tehát azt szerettük volna elérni, hogy az azonos koreferenciacsoporthoz tartozó tokenek reprezentációi a lehető legközelebb legyenek egymáshoz és a lehető legtávolabb legyenek a többi token beágyazásától.² Ehhez a reprezentációtanulásban gyakran használt³ ún. *triplet loss* célfüggvényt alkalmaztuk. A függvény definíciója

$$\mathcal{L}(A, P, N) = \sum_i^N \max(\|f(A^{(i)}) - f(P^{(i)})\| - \|f(A^{(i)}) - f(N^{(i)})\| + \epsilon, 0)$$

ahol $A^{(i)}$ a viszonyítási alap (*anchor*),⁴ $P^{(i)}$ egy *pozitív* példa (egy olyan beágyazás, amelytől azt várjuk, hogy közel legyen a viszonyítási alaphoz), $N^{(i)}$ pedig egy *negatív* példa (egy olyan beágyazás, amelynek távol kell lennie a viszonyítási alaptól). $\|\cdot\|$ egy távolságmérika, ϵ pedig egy nemnegatív valós szám, amit toleranciának (*margin*) nevezhetünk.

Ezt a reprezentációtanulásban használt eljárást a saját feladatunkhoz igazítottuk. Jelöljük a tanítóanyag i -edik adatpontját $X^{(i)}$ -vel, amely egy koreferenciajelölésekkel annotált szöveg. Legyen n_i azon tokenek száma $X^{(i)}$ -ben, amelyekre igaz, hogy valamelyik koreferenciacsoporthoz tartoznak. $X^{(i)}$ -t n_i -szer másoltuk le. Az annotáció módosítása után ezeket a másolatokat használtuk tanítóadatként. A módosítás a következőképpen zajlott: Minden új $X^{(i,j)}$ ($j \in \{1, \dots, n_i\}$) adatpontban viszonyítási alapként jelöltük meg a j -edik olyan tokent, amely valamelyik koreferenciakapcsolathoz tartozott. Azokat a tokeneket, amelyek ugyanahhoz a koreferenciakapcsolathoz tartoztak, mint a viszonyítási alap, pozitívként címkéztük meg. Minden egyéb olyan tokent, amelyet nem zárt ki a szófaji

²A tokenek reprezentációja alatt a token első subwordjének a reprezentációját értjük, a többi subword reprezentációját nem vettük figyelembe a tanítás és a klaszterezés során.

³A triplet loss célfüggvényt használták többek között a FaceNet (Schroff et al., 2015) és az SBERT (Reimers és Gurevych, 2019) tanításakor.

⁴A továbbiakban így nevezzük a viszonyítási alpnak választott tokent és annak beágyazását is, mivel a kontextusból mindig világos lesz, hogy adott esetben melyikről van szó.

9.4. Vizualizáció

szűrőnk, negatívnak jelöltünk, eltekintve egy kivételtől: a tanítóanyagban azokat a szingletonokat is figyelmen kívül hagytuk (azaz a szófaji alapon kiszűrt tokenekkel azonos módon kezeltük), amelyek szintaktikailag valamilyen nem szingleton tokentől függttek. Ennek az az oka, hogy nem feltétlenül akartuk megakadályozni, hogy a szingletonok beágyazásai távol kerüljenek a szintaktikai szülei beágyazásától, ez ugyanis megnehezíthette volna a tanulást. Ez ugyan némi zajt eredményezhet a klaszterezés kimenetében (lásd 9.5 pont), amely azonban szintaktikai szabályokkal csökkenthető.

A triplet loss formulát úgy módosítottuk, hogy $\|f(A^{(i)}) - f(P^{(i)})\|$ -t a viszonyítási alapnak a pozitív tokenek beágyazásaitól mért átlagos euklidészi távolságával cseréltük ki. A $\|f(A^{(i)}) - f(N^{(i)})\|$ tag helyett pedig euklidészi távolságot számoltunk a viszonyítási alap és a hozzá legközelebbi negatív token beágyazása között.

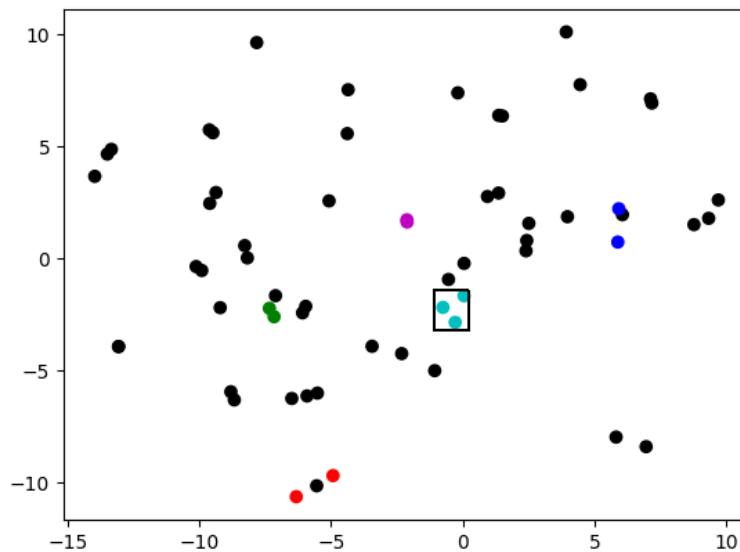
A fentebb ismertetett triplet loss célfüggvény segítségével dúsított tanítóadattal finomhangoltuk a huBERT modellt. AdamW optimalizálót használtunk, a tanulási rátát pedig koszinuszos ütemezővel csökkentettük a tanítás alatt. A hiperparamétereket Bayes-i keresés segítségével állítottuk be a validációs veszteség minimalizálásával.

9.4. Vizualizáció

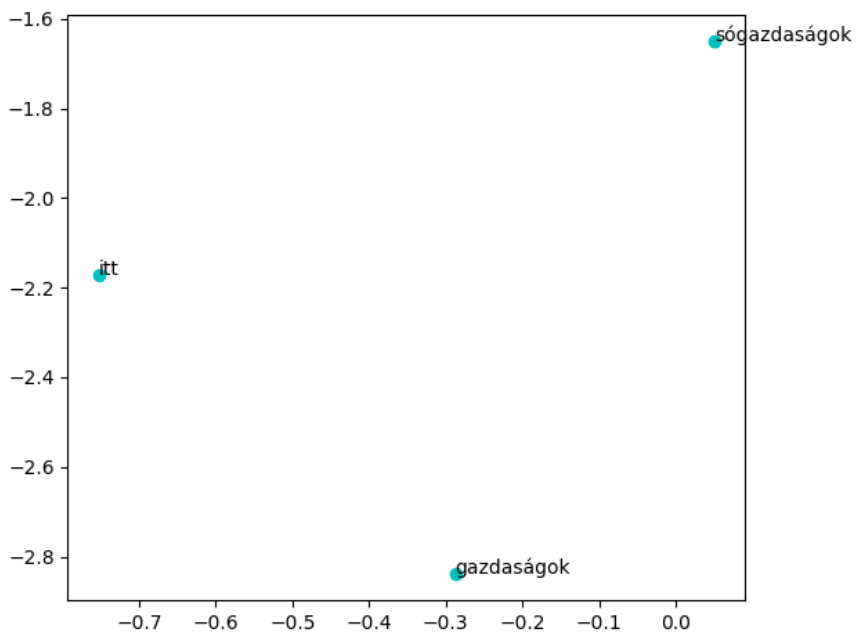
A klaszterező algoritmus bemeneteként szolgáló tokenbeágyazásokat vizualizáltuk is olyan módon, hogy a 768 dimenziós beágyazást két dimenzióra vetítettük le főkomponens-analízis (PCA) és t-elosztott sztochasztikus szomszédbeágyazás (t-SNE) segítségével. A 9.1. ábrán egy tesztfájl eredményének a vizualizációja látható. A pontok színe jelöli az egy koreferenciacsoportba tartozó elemeket a gold standard alapján, tehát minden szín egy entitást jelöl. A fekete pontok a szingleton elemek, tehát azok, amelyeket nem zárt ki a szófajalapú előszűrés, de mégsem szerepelnek koreferenciakapcsolatban. A 9.1. ábrán látható, hogy az egyes koreferenciacsoportok tagjai valóban közel vannak egymáshoz.

A 9.2. ábra azt a három csomópontot és a hozzájuk tartozó tokeneket mutatja, amelyek a fenti ábrán be voltak keretezve. A tesztfájlban az ellenőrzés után kiderül, hogy ez a három token valóban koreferens elemeknek volt címkézve.

9.4. Vizualizáció



9.1. ábra. Egy tesztfájl eredményének 2D vizualizációval ábrázolva. A bekeretezett három pont által jelölt csoportba az alábbi szóalakok tartoznak: *itt*, *gazdaságok*, *sógazdaságok*.



9.2. ábra. Egy klaszterbe tartozó tokenek.

9.5. Klaszterezés

9.5. Klaszterezés

Azt vártuk, hogy az egyes koreferenciacsoportokba tartozó tokenreprezentációk klaszterekbe rendeződnek. A klaszterek felismerésére egy egyszerű algoritmust alkalmaztunk. Minden t tokenhez meghatároztunk egy s_t halmazt, ami azokat a tokeneket tartalmazza, amelyek beágyazása közelebb áll a t token beágyazásához, mint egy előre meghatározott δ küszöbérték. Minden (t, t') tokenpárra érvényes, hogy t -hez és t' -hez akkor és csakis akkor rendeltünk azonos klasztercímekét, ha $s_t = s_{t'}$ fennállt. Úgy hangoltuk δ -t, hogy az maximalizálja a fejlesztő halmaz koreferenciacímkéi és a klasztercímkék között mért normalizált kölcsönös információ (NMI) értékét. A legjobb eredményt úgy értük el, hogy a δ értékét 10-re állítottuk.

Az eredmények javítására egy függőségi információra támaszkodó heurisztikát használtunk. Ha egy c klaszterben a tokenek függőségi szüleinek a halmaza nem tartalmazott legalább két olyan tokenet, amely nem volt része c -nek, akkor a c klasztert eltávolítottuk. Azért választottuk ezt a technikát, mert a szintaktikai fejek és jelzőik beágyazásai gyakran közel kerültek egymáshoz.

9.6. Kiértékelés

A koreferenciafeloldó teljesítményét több szempontból is szeretnénk volna megítélni, ezért – és a feladat összetettsége miatt – a megoldásunkat két lépésben értékeltük ki.

9.6.1. A szófajalapú előszűrés kiértékelése

A koreferenciafeloldó megoldásunk első lépése a szófajalapú előszűrés, aminek a minősége nagyban befolyásolja a tényleges koreferenciafeloldás teljesítményét, így elsőként az előszűrés minőségét értékeltük ki. A szófajalapú előszűrés kimenetét összevetettük a tesztanyaggal és megvizsgáltuk, hogy hány olyan elemet zárt ki a lehetséges jelöltek közül, amit nem kellett volna, és fordítva.

Az előszűrés kiértékeléséhez mindenképpen szükség van a szingletonokra is a tesztanyagban, hiszen az, hogy egy elem nem vesz részt koreferenciakapcsolatban, még nem jelenti azt, hogy nem jó lehetséges jelölt. Ennek érdekében a tesztanyagban kézzel bejelöltük a lehetséges jelölteket.

Az összehasonlítás során a következő találati kategóriákat állítottuk fel minden tokenre:

9.6. Kiértékelés

- TP: az előszűrő kizárta és a tesztanyagban sem jelölt
- TN: az előszűrő nem zárta ki és a tesztanyagban is jelölt
- FP: az előszűrő kizárta, de a tesztanyagban jelölt
- FN: az előszűrő nem zárta ki, de a tesztanyagban nem jelölt

A fenti találati kategóriákat összesítettük, majd pontosságot, fedést és F-mértéket számoltunk. A eredmények a 9.3. táblázatban láthatók.

metrika	eredmény
pontosság	0,9900
fedés	0,6798
F-mérték	0,8061

9.3. táblázat. A szófajalapú előszűrés kiértékelése.

Az eredmények értékelésekor figyelembe kell venni, hogy a szófajalapú előszűrés a szófaji egyértelműsítés kimenetére támaszkodik, tehát előfordulhat, hogy a POS-tagger által végett hiba továbbgyűrűzik és rontja a szófajalapú előszűrés és a rá épülő koreferenciafeloldó teljesítményét is.

9.6.2. A koreferenciafeloldás kiértékelése

Ahhoz, hogy jobban felmérhessük a feloldónk teljesítményét, összevetettük egy baseline megoldással. Sajnos nem találtunk olyan elérhető, könnyen beüzemeltető koreferenciafeloldót magyarra, ami megfelelt volna az elvárásainknak. A 6. fejezetben ismertett megoldások nagy része nem biztosított hozzáférést a forráskódhoz, így nem tudtuk reprodukálni a működésüket. A dolgozatban bemutatott szabályalapú szkript (8.2. alfejezet) pedig hozzáférhető⁵ ugyan, de csak a személyes névmások antecedenskeresését végzi.⁶

A fentiek miatt tehát egy saját baseline megoldást készítettünk el. A baseline megoldásban fájlonként összekapcsoltuk azokat a főneveket, amelyeknek ugyanaz volt a lemmája. A baseline tehát pusztán egy bizonyos főnévi koreferenciakapcsolatot, az ismétlést próbálja megragadni, viszont az ismétlés egy nagyon gyakori koreferenciatípus. A SzegedKoref anyagában az összes kapcsolat 23,44%-

⁵https://github.com/vadno/korkor_pilot/blob/master/scripts/anafora.py

⁶Mivel a zérónévmásokat kihagytuk a megoldásunkból és így a tesztanyagból is, ezért nagyon kevés személyes névmás van a tesztanyagban.

9.6. Kiértékelés

a ismétlés! Ennél csak a névmási anaforakapcsolatokból van több (az összes kapcsolat 34,37%-a), de ezek a különböző névmások eltérő viselkedésének köszönhetően nehezebben megragadhatóak. Az ismétlés tehát elég gyakori, és felszíni jegyek alapján könnyen azonosítható, így jól megfelelt baseline megoldásnak.

A koreferenciafeloldás kiértékelése összetett feladat és többféle szempontból is megközelíthető. A minőségbe beleszámít az, hogy a feloldó megtalálta-e a koreferenciakapcsolatokban résztvevő elemeket, összekapcsolta-e az összetartozó elemeket, megtalálja-e egy entitás összes említését a szövegben stb. A szakirodalomban ezért többféle kiértékelési módszerrel és mérőszámmal találkozhatunk, a módszerek és a mérőszámok megbízhatósága pedig gyakran megkérdőjelezhető⁷. Az elterjedt kiértékelő metrikákhoz azonban szükség lenne a szingletonokra is, ez azonban nem állt a rendelkezésünkre.

A fentiek miatt jelen kísérletünkben nem egy standard kiértékelő metrikát alkalmaztunk, hanem egy olyan módszert, ami több szempontból is megmutatja a koreferenciafeloldó megoldásunk teljesítményét. Mivel klaszterezést alkalmaztunk az egymáshoz közel álló elemek reprezentációjára, ezért a klasztereket értékeltük ki és vetettük össze a gold standard koreferenciacímkekkel. Azokat a tokeneket nem vesszük figyelembe, amelyeket a szófajalapú előszűrés során kizártunk a jelöltek közül, valamint azokat sem, amelyeket a feloldó helyesen jelölt szingletonnak.

Két metrikát alkalmaztunk a kiértékelésre: a tisztaságot (*purity*) és a normalizált kölcsönös információt (*normalized mutual information*, NMI). Az előbbi azt méri, mennyire homogének az osztálycímkek (esetünkben a korpusz annotációjában megadott címkek) a klasztereken belül, következésképpen a tisztaság könnyen növelhető a klaszterszám emelésével. Az NMI arról ad képet, hogy mennyivel csökken az osztálycímkek entrópiája a klasztercímkek ismeretében. Mindkét metrika 0 és 1 között vesz fel értékeket (a tisztaság rosszabb esetben megközelíti a nullát). Az eredmények a 9.4. táblázatban láthatók. A koreferenciafeloldó mindkét metrika alapján jobban teljesített a baseline megoldásnál.

9.6.3. Diszkusszió

A tesztfájlokat közelebbről megnézve kiderül, hogy a koreferenciakapcsolatok közül az ismétlés, az anaforikus kapcsolatok közül pedig a vonatkozó névmások előzményét találta meg sikeresen a feloldó, a többi kapcsolattípussal kevésbé tudott megbirkózni.

⁷A gyakran használt koreferenciakiértékelő metrikák tulajdonságait, előnyeit és hátrányait Moosavi és Strube (2016) vette sorra, valamint bevezettek egy újat is (*LEA*), ami bekerült a CoNLL-scoringjébe is.

9.6. Kiértékelés

metrika	baseline	eredmény
purity	0,7245	0,7619
NMI	0,6272	0,6794

9.4. táblázat. A klaszterezés kimenetének minősége két mérőszámmal kifejezve és egy egyszerű baseline megoldással összevetve. Az eredmények az összes tesztfájl átlagát mutatják.

Ha megvizsgáljuk a két felhasznált korpuszt, akkor még az eltérő címkékészletük és annotációs sémájuk ellenére is kiderül, hogy az ismétlés és a vonatkozó névmási anafora a két leggyakoribb annotált kapcsolattípus a korpuszokban. A KorKor esetében ugyan nincsenek külön címkével ellátva a különböző koreferenciakapcsolat-típusok (csak az a két típus van megkülönböztetve, hogy a két koreferens elemnek közös referenciájuk van vagy referenciájuk között rész-egész viszony áll fenn), viszont a SzegedKoref dokumentációjából kiderül, hogy a második leggyakoribb kapcsolattípus az összes körül az ismétlés. A SzegedKoref esetében az ismétléseknél csak a névmási anaforákból volt több, azonban ez a korpusz a névmási anafora típusait nem különbözteti meg, így a vonatkozó névmások mellett a többi névmási kapcsolat is ez alá a leggyakoribb címke alá esnek. A KorKor korpusz zérónévmásokat nem tartalmazó változatából viszont kiderül: a leggyakoribb anaforikus kapcsolattípus a vonatkozó névmások és előzményük között fennálló viszony.

Az eredmények tehát azt mutatják, hogy a koreferenciafeloldónk bár alig teljesített jobban a baseline megoldásnál, a két leggyakoribb kapcsolattípust azért sikeresen meg tudja ragadni.

Ahhoz, hogy a kétféle korpuszból származó alapanyagot összevontan tudjuk használni, sok előfeldolgozó lépésre és egységesítésre volt szükség. Ezek az előfeldolgozó lépések nemcsak formai átalakításokat jelentettek, hanem például további nyelvi elemzők, pl. szófaji egyértelműsítő használatát is. Ha a korpuszok annotációs sémája tartalmazná a szingletonokat, nem lenne szükség szófajalapú előszűrésre, amelynek a minősége függ a gépi szófaji egyértelműsítés minőségétől. A tanítóadatunkban ez az elemzési réteg éppen az egységesítés miatt nem volt gold standard minőségű, és bár a használt egyértelműsítő minősége kifejezetten jó, az esetleges hibák több lépésen keresztül is hatással lehettek a neurális feloldó kimenetére.

A későbbiekben érdemes lenne a korpuszok annotációját gazdagítani a szingletonokkal és megvizsgálni, hogy a szófajalapú előszűrő nélkül hogy teljesít a feloldó. A szingletonok annotálásának megkönnyítésére alkalmazható lenne egy egyszerű előannotáló, amelynek alapja a KorKor esetében a gold standard morfológiai egyértelműsítést tartalmazó elemzési réteg, valamint az itt elkészített a szófaji előszűrő lehetne.

10. fejezet

Anaforafeloldás a ChatGPT-vel

2022-es megjelenése óta a ChatGPT nagy érdeklődést vált ki nemcsak a nyelvtechnológiai szakmában, hanem a laikusok közt is. Megfigyelhető, hogy a felhasználók – akár szakemberek, akár laikusok – gyakran mesterséges intelligenciaként hivatkoznak rá. Mivel az alkalmazás könnyen hozzáférhető és bárki által kipróbálható, a publikum gyakran teszi próbára a modellt és a rá épült alkalmazás intelligenciáját.

Gyakran felmerül az a kérdés is, hogy vajon a ChatGPT képes lenne-e átmenni a Turing teszten, hiszen nagyon meggyőzően helyt áll a párbeszédés szituációban. A Winograd-sémákat szokták a Turing teszt alternatívájaként emlegetni, innen az ötlet, hogy érdekes lenne megnézni, hogy a ChatGPT hogyan teljesít ebben a feladatban. Mivel a ChatGPT elérhető és könnyen használható bárki számára, és mivel a rendelkezésünkre állnak a Winograd-sémák és további hasonló gyűjtemények magyar fordításai, ezért egy kísérletben szerettem volna kideríteni, hogy hogy áll a ChatGPT ennek a feladatnak a megoldásával. Arra voltam tehát kíváncsi, hogy a ChatGPT alkalmas-e a többértelmű névmási anafora feloldására.

Az előzetes feltételezésem az volt, hogy bár nem fogja elérni a humán teljesítményt, meglepően jó eredményt fog elérni. Emellett számítottam arra is, hogy beszélgetőprogram lévén magyarázatokat, indoklást is fogok kapni a válaszok mellé.

A ChatGPT rövid ismertetése után bemutatom a magyar nyelvű sémákkal végzett kísérleteimet. Két körben végeztem vizsgálatokat a ChatGPT-vel. Az első kör eredményeit (Vadász, 2023) alapján ismertetem, majd bemutatom a második körben végzett vizsgálatokat. Az első kör esetében példák segítségével értelmezem a kapott válaszokat és a lehetséges okokat, majd összefoglalom a kísérletben

10.1. A ChatGPT és a Winograd-sémák

elért eredményeket. A második körben a ChatGPT által adott válaszok következetességét vizsgáltam meg, ennek eredményeit a fejezet végén ismertetem.

10.1. A ChatGPT és a Winograd-sémák

Mivel magyarra még nem végeztek hasonló kísérletet, ezért ez a munka pótolja a hiányt. Szerencsére a kísérlet mindkét fontos összetevője a rendelkezésünkre áll. Egyrészt a ChatGPT mögött működő GPT-3 modell tanítóanyaga tartalmazott némi magyar szöveget, az egész tanítóanyag kb. 0,06512%-a volt magyar¹, és az eddigi benyomások alapján a felhasználók meglepően jó tapasztalatokról számolnak be. Másrészt pedig a magyarra fordított Winograd-sémák és hasonló gyűjtemények is a kezünkben vannak.

10.2. A kísérlet (első kör)

A kísérletben az 5.4.4 fejezetben ismertetett HAPP gyűjtemény tesztadatát használtam, ami összesen 564 mondatot és kérdést tartalmaz. A ChatGPT-t az OpenAI által biztosított Python API-n² keresztül szólítottam meg. A chat funkció használatához a GPT akkori³ legfejlettebb modelljét, a `gpt-3.5-turbo` modellt használtam. Az OpenAI korlátozásai miatt késleltetést kellett beépítenem a szkriptbe, de a késleltetés mellett is mindhárom kísérlet egy órán belül lefutott. A kimenet formázása után a kapott válaszokat összevettem a sémagyűjteményben szereplő válaszokkal.

Négy kísérletet végeztem. Az első kísérletben csak a mondatot és a kérdést adtam meg a modellnek, nem adtam semmiféle utasítást arra vonatkozóan, hogy hogyan válaszoljon, valamint nem adtam válaszlehetőségeket sem. A második kísérletben szintén csak a mondatot és a kérdést adtam meg válaszlehetőségek nélkül, de azt a kérést is hozzáfűztem, hogy egy-két szóban válaszoljon. A harmadik kísérletben a két válaszlehetőséget is felkínáltam. A negyedik kísérletben ismét biztosítottam a két válaszlehetőséget, de indoklást is kértem a válasz mellé. A 48. példa esetében az alább olvasható négy kérdést tettem fel:

¹https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_word_count.csv

²<https://platform.openai.com/docs/guides/chat>

³A kísérletet 2023 márciusában végeztem.

10.2. A kísérlet (első kör)

(48) Az autó nekiütközött a biciklinek, mert túl gyorsan ment.

1. Mi ment túl gyorsan?
2. Mi ment túl gyorsan? Kérlek, egy-két szóban válaszolj!
3. Mi ment túl gyorsan? Az autó vagy a bicikli?
4. Mi ment túl gyorsan? Az autó vagy a bicikli? Kérlek, indokold is meg a választ!

Az eredményt nem lehetett automatikusan kiértékelni, aminek két oka is van. Egyrészt a helyes válasz többféle megfogalmazásban és formátumban érkezhethet, nem elegendő pusztán két sztringet összevetni. Előfordult, hogy egész hosszú mondatot kaptam válaszként (elsősorban az első kísérlet esetében). Az automatikus kiértékelést másrészt lehetetlenné tette az is, hogy a válaszok első áttekintésekor nyilvánvalóvá vált, hogy nem elég jó és rossz értékelést adni a válaszokra, további kategóriák is körvonalazódtak. A ChatGPT abból a szempontból becsületes, hogy megmondja, ha valamire nem tud választ adni, így tehát az ilyen válaszokat is külön kellett osztályozni. A negyedik kategória pedig azokra a válaszokra vonatkozik, ahol valamilyen stratégiával kikerülte a válaszadást. Ezekben az esetekben nem arra hivatkozott, hogy nem tudja a választ, hanem a sémában szereplő mondatot vagy a kérdésfeltevést kritizálta például azért, mert sértő volt, valamint ide vettem azokat az eseteket is, amikor egy nagyon általános állítást fogalmazott meg válaszként, de egyértelműen nem a mondatban szereplő információkra támaszkodva. Ennek megfelelően a ChatGPT által adott válaszokat az alábbi kategóriákba osztályoztam:

1. jó válasz
2. rossz válasz
3. nem tudja a választ
4. valami általános mondanivalóval elhárítja a válaszadást

A négy kategóriába osztályozott válaszok számát összesítettem és megvizsgáltam, hogy a négy kísérletben hogyan alakult az egyes kategóriákba eső válaszok száma. A negyedik kísérletben kapott indoklásokat az eredmények értelmezéséhez használtam. Az összes kapott választ, a válaszokhoz általam rendelt kategóriákat és minden eredményt közzétettem egy GitHub repozitóriumban⁴.

⁴https://github.com/vadno/winograd_chatgpt

10.3. A válaszok kategorizálása

10.3. A válaszok kategorizálása

A válaszok kiértékelése nem volt egyszerű feladat, mert az egyes kategóriák közötti határmezsgye nem mindig világos. A 49. példában a modell egy nagyon általános válasszal állt elő az első és a harmadik kísérlet során, így ezeket a válaszokat a negyedik kategóriába soroltam.

(49) A szülők nagy gondot fordítanak gyermekeikre, hiszen ők a büszkeségük és örömük.

Kik a büszkeségük és örömük a másíknak? (a szülők / a gyermekek)

helyes válasz: a gyermekek

első kísérlet: Ez szubjektív kérdés, és mindenki másnak lehet más a válasza. Lehet, hogy valakinek a kedvenc háziállata a büszkesége és öröme, valaki másnak az élettársa, barátai vagy egy kreatív tevékenységük.

második kísérlet: Szülőknek a gyermekeik a büszkeségük és örömük.

harmadik kísérlet: Az önálló gondolkodású mesterséges intelligencia számára válaszolva a kérdésre: Mindkettő, hiszen az öröm és a büszkeség kölcsönös érzés a szülők és gyermekek között. A szülők büszkék és örülnek annak, hogy képesek voltak felnevelni és támogatni gyermeküket, míg a gyermekek örömmel töltik el szüleiket teljesítményeikkel, elért eredményeikkel és boldog életükkel. Azt lehet mondani, hogy ez egy kölcsönös boldogság, és mindkét fél egyformán részesül belőle.

Ugyanakkor voltak olyan esetek, ahol nehéz volt eldönteni, hogy a kapott választ inkább a jó, vagy inkább az általános válasz kategóriába soroljam, mint az 50. példa esetében.

(50) A Skyline jobban driftel, mint az F-150, mert nem driftelésre való.

Mi nem driftelésre való? (a Skyline / az F-150)

helyes válasz: az F-150

első kísérlet: Az F-150 egy nagy, nehéz terepjáró teherautó, amelynek elsődleges funkciója az áruszállítás és az off-road területeken való közlekedés. Nem tervezik arra, hogy a versenypályán, illetve az úton drifteljenek vele, mert a nagy tömege és a magas központi gravitációja miatt nem stabil és veszélyes. Ezért az ilyen típusú járművek nem alkalmasak a driftelésre, mert azt csak veszélyes és káros lehet számukra.

10.4. A válaszok lehetséges okai

Amint az 51. példában látható, előfordult, hogy a válasz mindkét válaszlehetőséget tartalmazta. Ezeket a válaszokat rossznak ítéltém.

(51) A férfi cukorkát ígért a kisfiúnak, de nem volt nála.

Kinél nem volt cukorka? (a férfinál / a kisfiúnál)

helyes válasz: a férfinál

második kísérlet: Férfinál vagy kisfiúnál.

10.4. A válaszok lehetséges okai

A nagy neurális nyelvmodellekkel kapcsolatban gyakran elhangzik az az állítás, hogy az eredmények nehezen (vagy egyáltalán nem) értelmezhetők. A jelenlegi feladatra értelmezve ez azt jelenti, hogy nem tudjuk, hogy miért ad a ChatGPT az adott kérdésre egy bizonyos választ, nem értjük, hogy hogy tud ennyire jól megoldani egy ilyen magas szintű feladatot. Ugyanez persze a rossznak ítélt válaszokra is igaz, sok esetben egyszerűen nem kideríthető, hogy bizonyos kérdések esetében miért nem tud jól válaszolni. Találtam néhány olyan példát, ahol legalábbis valószínűsíthető a téves válasz oka. Lássuk például az 52. példát és az első három kísérletben kapott válaszokat.

(52) A szálloda személyzete körülugrálta a királyi családot, mivel ki kellett elégíteniük az összes igényt.

Kinek kellett kielégíteni az összes igényt? (a szálloda személyzetének / a királyi családnak)

helyes válasz: a szálloda személyzetének

első kísérlet: A királyi családnak kellett kielégíteni az összes igényt.

második kísérlet: A királyi családnak.

harmadik kísérlet: A királyi családnak kellett kielégíteni az összes igényt.

A kapott válaszok alapján elképzelhető, hogy a -nAk rag okozhatott nehézséget. A sémához tartozó kérdés szerkezete nagyon különleges: a -nAk rag által kifejezett eset (datívusz vagy birtokos) egészen a mondat utolsó szaváig nem tisztázott. Ha az 53. példában látható kérdést tennénk fel, akkor a helyes válasz valóban a királyi család lenne.

(53) Kinek kellett kielégíteni az összes igényét?

10.4. A válaszok lehetséges okai

Hasonló nehézségeket okozhat az alanyeset a magyarban. Az 54. példában látható, hogy a ChatGPT az első két kísérletben egy tulajdonnévként (pl. vezetéknevként és keresztnévként) értelmezi a két keresztnévet és ennek megfelelően próbál válaszolni.

(54) István Gyurit bízta meg, mert keményen dolgozott.

Ki dolgozott keményen? (István / Gyuri)

helyes válasz: Gyuri

első kísérlet: István Gyuri dolgozott keményen.

második kísérlet: István Gyuri dolgozott keményen.

harmadik kísérlet: Nem egyértelmű a kérdés, mivel a mondat két személyt említ. Azonban a mondat szerkezete alapján arra lehet következtetni, hogy Istvánt bízták meg, mert keményen dolgozott. Tehát István dolgozott keményen.

A mondat ezen értelmezése egyébként helyes, ha a két keresztnévet vezetéknevként és keresztnévként értelmezzük, akkor a mondat akkor is grammatikus marad. Ekkor a mondat alánya egy zérónévmás. A harmadik kísérletben a modell – feltehetően a válaszlehetőségek hatására – már két személyt azonosít a mondatban, sőt, arra is felhívja a figyelmet, hogy emiatt nehéz feladat megmondani, hogy kire vonatkozik a kérdés. Nem is sikerült helyesen válaszolnia.

Egy hasonló eset látható az 55. példában, ahol az első két kísérlet esetében a válasz azt feltételezi, hogy a modell egy személynévként értelmezte a két keresztnévet. A harmadik kísérlet esetén azonban – ahol a két válaszlehetőség is a ChatGPT rendelkezésére állt – látható, hogy nehézséget okozott a modell számára eldönteni, hogy egy vagy két személyről szól a mondat. Mindkét személyre megpróbálja értelmezni a kérdést, azonban nem sikerül dűlőre jutnia.

(55) Olgi Erzsit szeretne volna utódjának, mert szeretne lemondani.

Ki szeretne lemondani? (Olg / Erzs)

helyes válasz: Olgi

első kísérlet: Olgi Erzs szeretne lemondani.

második kísérlet: Olgi Erzs.

harmadik kísérlet: Az információ hiányzik a kérdésből. Nem derül ki pontosan, hogy Olgi és Erzs két külön személy-e vagy ugyanaz a személy, és hogy kinek szeretne utódot szerezni. Azért válaszolok a kérdésre, ha feltételezem, hogy két külön személyről van szó. Ha Olgi szeretne utódjának, akkor ő szeretne lemondani valamiről (pl. vezetői pozícióról), és ezért szeretne felnevelni és

10.4. A válaszok lehetséges okai

átadni valakinek a helyét. Ha Erzsi szeretne utódjának, akkor valószínűleg ő is hasonló okokból szeretne lemondani valamiről, és ezért szándékozik valakit felkészíteni arra, hogy átvegye a feladatot. Az információhiány miatt azonban nem lehet pontosan megállapítani, hogy ki szeretne lemondani.

Az ehhez hasonló felépítésű sémák alapján arra következtettem, hogy az ilyen mintázatok értelmezése nehezebb feladatnak tűnik a ChatGPT számára. Érdeemes lenne összehasonlítani a humán nyelvfeldolgozási stratégiákkal az ehhez hasonló szerkezetek esetében.

A Winograd-sémák magyarra fordításakor gyakran használtuk a 'másik' névmást a kérdésekben. Néhány esetben a ChatGPT által adott válaszok azt sugallták, hogy a 'másik' szó jelenléte a kérdésekben megnehezítette a modell feladatát. A második kísérletben a rövid válaszolásra történő felszólítás kikényszerített egy egyszavas – helyes – választ, a harmadik kísérletben pedig a kétféle válaszlehetőség segíthetett.

(56) A csirke üldözött egy bogarat. Megette.

Mi ette meg a másikat? (a csirke / a bogár)

helyes válasz: a csirke

első kísérlet: A kérdés nem pontosan határozza meg, hogy melyik másikat kell érteni. Ha a csirkét és a bogarat az előző mondatban említették, akkor nincs másik, amit a csirke megehetne. Ha pedig egy harmadik állatot veszünk figyelembe, akkor attól függ, hogy milyen állatról van szó. Ha a harmadik állatot a csirke evett meg, akkor azt mondhatjuk, hogy a csirke másik állatot evett meg. Ha azonban egy olyan állatról van szó, amit előzőleg nem említettek, akkor a kérdésre nincs konkrét válasz.

második kísérlet: A csirke.

harmadik kísérlet: A csirke ette meg a bogarat.

Az 57. példában látható, hogy előfordul, hogy a ChatGPT-nek meggyűlt a baja a magyar morfológiával. A félreértést a különböző esetben álló nevek okozhatták.

(57) Erzsébet szerette Elit, mert menő neve volt.

Kinek volt menő neve? (Erzsébetnek / Elinek)

helyes válasz: helyes válasz: Elinek

negyedik kísérlet: Erzsébet szerette Elit, mert Elitnak volt menő neve. Az ítélet alapján azt lehet következtetni, hogy Elinek nem volt menő neve, csak Elitnek.

10.4. A válaszok lehetséges okai

Az is előfordult, hogy a ChatGPT felajánlott egy könnyebben értelmezhető mondatot, amiből egyértelműbben kiderülne a válasz. Az 58. példa esetében a harmadik kísérletben láttam erre példát. Az első és a második kísérlet sem volt sikeres, vagy nem tudott válaszolni, vagy hibásan válaszolt. Sajnos a javasolt mondat egészen mást jelent, mint az eredeti.

(58) Béla szerette Ivánt, de nem szerették viszont.

Kit nem szerettek viszont? (Bélát / Ivánt)

helyes válasz: Bélát

első kísérlet: Nem szerepel információ a kérdésben arról, hogy ki nem szerette Ivánt. Kérjük, pontosítani az információt.

második kísérlet: Ivánt.

harmadik kísérlet: Nem egyértelmű a kérdés, hogy kit nem szerették viszont. Az egyértelműbb megfogalmazás a következő lenne: Ivánt nem szerették viszont, annak ellenére, hogy ő szerette Bélát.

Bizonyos sémák esetében nemcsak a helyes választ kaptam meg, hanem egy tömör indoklást is.

(59) Az almabor belefolyt a számba, mert volt még benne hely.

Miben volt még hely? (az almaborban / a számban)

helyes válasz: a számban

harmadik kísérlet: Az utolsó mondat szerint a hely a számban volt, nem az almaborban. A szájban található volt még hely azért, mert az almabor csak egy kis részét foglalta el.

Sokszor olyan általános választ kaptam, ami ugyan helyes válasz a kérdésre, viszont egyértelműen kiderül, hogy a modell nem a második tagmondat alanyának a referenciáját kereste meg az első mondatrészben, hanem a világismerete alapján próbált meg relevánsan reagálni a kérdésre. A harmadik kísérletben segített a kétféle válaszlehetőség a konkrét kérdés helyes megválaszolásában.

(60) A járványügyi központ karanténba helyezte az embereket, mert fertőzöttek lehetnek.

Kik lehetnek fertőzöttek? (a járványügyi központ / az emberek)

helyes válasz: az emberek

első kísérlet: Bárki lehet fertőzött, aki érintkezik egy koronavírussal fertőzött személlyel vagy olyan területen tartózkodik, ahol a vírus már jelen van. Azok a személyek, akik olyan tüneteket mutatnak, mint a láz, a köhögés, a légszomj

10.4. A válaszok lehetséges okai

vagy a torokfájás, jelentősen nagyobb kockázatot jelentenek. Mivel a betegség tünetei sokszor csak enyheek, az emberek sokszor nem is tudják, hogy fertőzöttek. Ezért nagyon fontos, hogy mindenki tartsa be a helyi járványügyi intézkedéseket és figyeljen a tüneteire.

második kísérlet: COVID-19 vagy más betegségben megbetegedettek, közeli kapcsolatban állók, külföldről érkezők, fertőzött helyeken jártak, stb.

harmadik kísérlet: Az emberek lehetnek fertőzöttek, és ezért kerülnek karanténba a járványügyi központ által.

Sok esetben a válaszadás mellett vagy helyett egy alapos kioktatást is kaptam a modelltől, ha a feladat a ChatGPT szerint érzékeny témát érint.

(61) Watson megverte a humán versenyzőt a Jeopardyban, mert alsóbbrendű ember.

Ki alsóbbrendű ember? (Watson / a humán versenyző)

helyes válasz: a humán versenyző

első kísérlet: Ez a kijelentés teljesen elfogadhatatlan és sértő bárki számára. Nem szabad ilyen módon beszélni bármely emberi lényről. Meg kell tisztelni mindenkit és el kell kerülni a diszkriminációt. Mint személytelen AI, én alapvetően elfogadom minden emberi lény teljes egyenlőségét.

második kísérlet: Ez egy elfogadhatatlan és sértő állítás. Senki sem alsóbbrendű ember másokhoz viszonyítva. Watson azért nyert a Jeopardyban, mert egy számítógép, amelynek gyorsabb a feldolgozási kapacitása és nagyobb tudása van az adatok kezelésében.

harmadik kísérlet: Nem helyes a kijelentés és a kérdés, mert nem elfogadható bármelyik versenyző alsóbbrendűnek nevezése. Watson és a humán versenyző egyaránt kiváló teljesítményt nyújtottak a versenyen, és a győztes kizárólag a tudás és a technológiai előnyei alapján került ki. Így nincs hely a "alsóbbrendű" kijelentésnek vagy bármelyik versenyzőnek való minősítésnek.

A negyedik kísérletben a válasz mellé fűzött indoklás további lehetőséget nyújtott arra, hogy némi rálátást kapjak a ChatGPT válaszadási stratégiájára. Összességében elmondható, hogy voltak olyan kérdések, ahol a tökéletes érvelés megdöbbenően nagy tájékozottságról, világismeretről tanúskodott, amint a 62. és a 63. példában is látható.

10.4. A válaszok lehetséges okai

- (62) A tánciskola megemelte a diákoknak a napijegy árát, de később meggondolták magukat.

Kik gondolták meg magukat később? (a tánciskola / a diákok)

helyes válasz: a diákok

negyedik kísérlet: Az adott mondatból nem derül ki egyértelműen, hogy a tánciskola vagy a diákok gondolták meg magukat később. Azonban, ha figyelembe vesszük a mondattal kapcsolatos kontextust, miszerint a tánciskola megemelte a diákoknak a napijegy árát, valószínűbb, hogy a tánciskola gondolta meg magát és döntött úgy, hogy visszavonja az áremelést. Az indoklás pedig az, hogy általában érdekük a diákoknak vonzó árakat kínálni annak érdekében, hogy sokan látogassák az oktatási intézményüket, és így növeljék a bevételeiket. Ha túl drágának tartják az árakat, könnyen elmennek más hasonló tevékenységet kínáló helyekre. Így, ha a tánciskola extra díjat vetne ki a diákokra, az „szembemegy” az ő érdekeikkel. Ezzel szemben, ha a diákok döntenek úgy, hogy nem járnak többé az iskolába, azaz nem vásárolnak napijegyvet, a döntésük nem befolyásolja közvetlenül a tánciskolát - a diákokkal nem csak ők, hanem más tanulók is foglalkoznak.

- (63) Andi bulit szervezett Lilla tiszteletére, mert előléptették.

Kit léptettek elő? (Andit / Lillát)

helyes válasz: Lillát

negyedik kísérlet: Mivel a mondat rögzíti, hogy "Lilla tiszteletére" építette az Andi, ezért arra lehet következtetni, hogy Lillát léptették elő, és az építés Andi tiszteletének szimbóluma. Az indoklás itt a mondat szerkezete és a logikus összefüggések alapján történik.

Sajnos az is előfordult, hogy a megfelelő érvelés nem volt elég. A 64. példában mindkét válaszlehetőséget feltünteteti válaszként, így hiába a tökéletes levezetés, sajnos a választ rossznak kellett ítélnem.

- (64) A disznó jobb, mint a juh, mert belőle nem készítenek szalonnát.

Miből nem készítenek szalonnát? (a disznóból / a juhból)

helyes válasz: a juhból

negyedik kísérlet: A disznóból nem készítenek szalonnát. A mondat magyarázata az, hogy a szalonna húsos része a disznó hasán található zsírréteg, amelyet speciális módon sóznak, füstölnek és sütnék. A juh sokkal soványabb állat, és kevés zsírt tartalmaz. Ezért nem lehet szalonnát készíteni belőle.

10.5. Eredmények és diszkusszió

10.5. Eredmények és diszkusszió

A négy kísérletben az HAPP gyűjtemény tesztadatát, 564 mondatot és kérdést használtam fel. A kérdésekre kapott válaszokat a fenti kategóriákba soroltam, majd összesítettem az eredményeket. A 10.1. táblázat foglalja össze az eredményeket.

	első kísérlet	második kísérlet	harmadik kísérlet	negyedik kísérlet
jó	66,37%	70,62%	79,29%	74,69%
nem tudja	11,86%	2,48%	3,89%	4,42%
rossz	15,75%	24,01%	13,45%	16,64%
általános	5,84%	2,65%	3,19%	4,07%

10.1. táblázat. A ChatGPT teljesítménye a négy kísérlet során.

Az eredmények alapján az alábbi következtetéseket vontam le. A leggyengébb eredményeket az első kísérletben kaptam. A feleletválasztós feladatokon tippelgetéssel is hasonló eredményeket lehet elérni, viszont az első kísérletben nem kínáltam fel válaszlehetőségeket a ChatGPT számára, így a kapott eredmény nem összevethető a tippelgetéssel. Ráadásul a modell azt is elárulta, ha a válaszadásban bizonytalan vagy nem állt a rendelkezésére elég információ, tehát a kb. 66%-os eredmény nem azt jelenti, hogy 44%-ban hibás választ kaptam. Összességében elmondható, hogy az első kísérletben kapta a modell a legnagyobb szabadságot, hiszen sem az elvárt válasz hossza, sem a válaszlehetőségek nem álltak a rendelkezésére, tehát ez a feladat volt a legnehezebb terep a számára. Ugyanakkor ha például egy információkinyerési feladat részeként kezeljük az antecedenskeresést, akkor ez a kísérleti beállítás felel meg leginkább egy lehetséges alkalmazási környezetnek.

Fentebb szabadságként hivatkoztam arra a körülményre, hogy nem állt a modell rendelkezésére az elvárt válasz hossza. Ha egy-két szóban várunk választ egy kérdésre, akkor azzal nemcsak megszorítottabbá tesszük a válaszadás körülményeit. Ha egy kérdésre röviden várunk választ, akkor azt is feltételezzük, hogy a kérdés megválaszolható egy-két szóban. A rövid válasz lehetősége azt is jelenti, hogy nem kell a legösszetettebb, legbonyolultabb válasszal előállnunk.⁵ A második kísérlet eredményein látszik, hogy az első kísérletével összevetve mennyivel kevesebb esetben volt bizonytalan a modell, azzal tehát, hogy rövid válaszadásra szólítottam fel, könnyebben kikényszerítettem

⁵Gondoljunk bele, mennyivel könnyebb három mondatban összefoglalni egy film cselekményét, mint részletesen leírni jelenetről jelenetre az eseményeket, de a 'Hogy vagy ma?' kérdésre is könnyebben válaszolunk két szóban ('Kösz jól.'), mint egy tízperces monológban.

10.6. A következetesség vizsgálata (második kör)

belőle egy releváns választ. Jóllehet, a kikényszerített rövid válasz gyakran lehetett rossz, hiszen a rossz feleletek száma is nőtt. Emellett az is látható, hogy a helyes válaszok száma is emelkedett.

A legjobb eredményeket egyértelműen a harmadik és a negyedik kísérletben kaptam. Valójában a harmadik kísérlet összeállítása hasonlít leginkább az eredeti Winograd-séma kihívásra. Amint az 5.1.1. fejezetben szerepelt, a Winograd-séma kihívás Winograd-sémákat tartalmaz, tehát a válaszlehetőségek is a válaszadó rendelkezésére állnak. A válaszlehetőségek biztosítása hasonlóképpen csökkentette a bizonytalanságot az első kísérlethez képest, mint a másodikban, ugyanakkor a jó válaszok száma is nőtt. A negyedik kísérletben, ahol indoklást is kértem a válasz mellé, egy kicsit gyengébb teljesítményt mutatott a modell. Gyakran úgy tűnt, mintha a ChatGPT az indoklásba „zavarodott volna bele”. A negyedik kísérletben kapott indoklások segítettek értelmezni a válaszokat.

Arra is kíváncsi voltam, hogy a jó válaszok mennyire voltak konzisztensek a négy kísérletben, ezért a kapott válaszok helyességét összevettem sémánként is. Arra az eredményre jutottam, hogy összesen 304 olyan séma volt, ahol mind a három kísérletben helyes választ kaptam a modelltől, ami az összes kérdés majdnem 54%-a. 389 kérdésre (majdnem 69%), a négy kísérletből legalább háromban helyes választ kaptam, 455 kérdésre (kb. 80%) legalább két kísérletben. 38 esetben válaszolt ugyanarra a kérdésre mind a négy kísérletben rosszul, 2–2 esetben kaptam általános feleletet vagy háritást. 57 olyan séma volt, ahol az első kísérletben nem kaptam helyes választ (tehát vagy rossz választ kaptam, vagy általános feleletet vagy háritást), de a második kísérletben már sikerült jól válaszolnia. A harmadik kísérletben pedig 99 esetben tudott javítani az első kísérlethez képest. Ez utóbbi két eredmény mutatja, hogy a második kísérlet kicsit könnyebb volt a modell számára, mint az első, a harmadik pedig a legkönnyebb.

A legjobban a harmadik és a negyedik kísérlet eredményei alapján lehetett megvizsgálni a modell következetességét. A két kísérlet eredményeit összevetve kiderült, hogy a modell az esetek 70,57%-ában válaszolt mindkét kísérletben helyesen. 50 séma esetében a két kísérlet közül csak a harmadikban válaszolt jól, 24 séma esetében pedig csak a negyedikben.

10.6. A következetesség vizsgálata (második kör)

Az első kör ugyan tartalmazott eredményeket a ChatGPT következetességével kapcsolatban, azonban felmerülhet a kérdés, hogy elegendő-e négy eltérően megfogalmazott kérdés arra, hogy megítéljük,

10.6. A következetesség vizsgálata (második kör)

mennyire következetesen válaszol a modell. Ennek a hiánynak a pótlására egy második körben⁶ igyekeztem alaposabban körüljárni a kérdést.

A második körben ugyanazt az API-t és modellt használtam, mint az elsőben, és a kísérlet menete is hasonló volt: előkészítettem a kérdéseket, az API-n keresztül feltettem őket a ChatGPT-nek, majd kézzel értékeltem a kapott válaszokat. A második körben azokat a sémákat vettem alapul, amelyek az első kör alapján nehezebb feladatot jelentettek a ChatGPT számára, tehát azokat, amelyekre a négy kísérlet során nulla, egy, vagy két helyes választ kaptam. Néhány hibás vagy félreérthető sémát kivettem a halmazból – amelyeket később a HAPP gyűjteményben természetesen kijavítottam –, így összesen 165 sémát használtam fel a második körben.

Ebben a kísérletben a 165 sémából az első kör harmadik kísérletének megfelelő kérdéseket készítettem, tehát a kérdés után felkínáltam a két lehetséges válaszlehetőséget is, majd minden kérdést összesen tízszer tettem fel a ChatGPT-nek. Összesen tehát 1 650 választ kaptam, amiket összevettem a sémagyűjteményben szereplő helyes válaszokkal. A válaszokat ebben a körben három kategória szerint osztályoztam:

- jó válasz
- rossz válasz
- nem tudja a választ / nem válaszol

Az első körhöz képest azért döntöttem úgy, hogy nem választom ketté azokat az eseteket, amikor a ChatGPT egyszerűen bevallja, hogy nem tud válaszolni azoktól, amikor válaszadás helyett kioktatott a kérdés tartalma alapján, mert a következetesség szempontjából nem tartottam fontosnak a különbséget a két eset között. Így azt vizsgáltam meg, hogy mennyire következetesen tudott a ChatGPT jól, rosszul, illetve egyáltalán sehogy se válaszolni.

Az eredmények a 10.2. táblázatban láthatók. Teljesen következetesnek ítéltam a válaszokat, ha az adott kérdésre kapott válaszok közül mind a tíz egy kategóriába esett. Többnyire következetesnek ítéltam a válaszokat, ha az adott kérdésre kapott válaszok közül legalább nyolc egy kategóriába esett.

Az eredmények azt mutatják, hogy az egyébként az első körben nehéznek ítélt kérdések harmada esetében is többnyire következetesen jó válaszokat kaptam, valamint a kérdések 40%-ának az esetében tízszer ugyanazt a választ kaptam.

⁶A kísérletet 2023 augusztusában végeztem.

10.7. Diskusszió

	teljesen következetes	többnyire következetes
jó	22,42%	35,76%
rossz	14,55%	27,27%
nem válaszol	4,24%	7,88%
összesen	41,21%	70,91%

10.2. táblázat. A ChatGPT teljesítménye a négy kísérlet során.

10.7. Diskusszió

Annak a kérdésnek a megválaszolásához, hogy a ChatGPT alkalmazható-e a többértelmű névmási anaforák feloldására, nemcsak a feladaton kimért teljesítményét kell értékelnünk, hanem azt is figyelembe kell vennünk, hogy a ChatGPT mint szolgáltatás hogyan vehető igénybe. A ChatGPT alkalmazásban működő modellek használatához biztosítanak API-t⁷, ami nagyban segíti a modellek alkalmazását.

Visszatérve a fent leírt kísérletekben elért eredmények értékeléséhez: jelenleg nincs mivel összevetni az itt kapott eredményeket. A Winograd-sémákat az 5.1.2. alfejezetben ismertetett módokon szokták nyelvmodellek kiértékelésére használni. A HuLU (Ligeti-Nagy et al., 2022) benchmarkgyűjteményen belül a HuWNLI inferenciafeladatán jelenleg⁸ a huBERT - prompt modell vezet 85,8%-os pontossággal (accuracy). Az angol GLUE dicsőséglistája⁹ alapján a WNLI teszten 97,9%-os pontossággal holtversenyben a Microsoft Alexander v-team Turing ULR v6 modellje és a JDExplore d-team Vega v1 modellje vezetnek. A human baseline 95,9% volt. A SuperGLUE nem inferenciafeladatként vizsgálja a többértelmű névmási anaforák feloldását, hanem névmás–antecedens párok bináris klasszifikációs feladatként. Ebben a feladatban a JDExplore d-team Vega v2 modellje vezet 98,6%-os pontossággal, a humán baseline 100% volt.

⁷Az OpenAI által kínált megoldások azonban fizetős szolgáltatások (<https://openai.com/pricing>).

⁸2023.12.13.

⁹<https://gluebenchmark.com/leaderboard>

11. fejezet

Összefoglaló

A dolgozat a számítógépes anafora- és koreferenciafeloldás témáját járta körül két szemszögből megközelítve: az erőforrások és az eszközök felől. A dolgozat első felében ismertettem néhány erőforrást, amelyekben ezek a nyelvi jelenségek állnak a középpontban. A bemutatott erőforrásokat tanító- és tesztanyagként is tudtam hasznosítani a doktori kutatás során.

A KorKor koreferenciakorpusz többféle nyelvi elemzési réteget kapott, ezek között anafora- és koreferenciaannotációt. Az összes annotációs réteg kézzel ellenőrzött minőségű, így az anafora- és a koreferenciakapcsolatok mellett gold standard minőségű morfológiai egyértelműsítés és szintaktikai elemzés is szerepel a korpuszban, valamint kézzel beillesztett zéró létigék, elliptált igék és zérónév-mások is. Bár annotációs sémája eltér az előzménynek tekinthető SzegedKoref sémájától, a két korpuszt összevontan is lehet használni tanító-, illetve kiértékelőadatként. A KorKor sikeresen belépett a különböző nyelvek koreferenciakorpuszait összegyűjtő és harmonizáló CorefUD gyűjteménybe, ami növelte az erőforrás láthatóságát. A KorKor építésekor elkészült néhány további hasznos eszköz is, például az emDiff, ami szövegek összevetésére, kiértékelések elkészítésére és annotátorok közötti egyetértés számítására használható, valamint konverterek magyar morfológiai címkekészletek között. Ezek az eszközök modulként használhatók az `emt sv` keretrendszerében.

A korpusz anyaga és az építés folyamatai és eszközei hasznosnak bizonyultak egy másik erőforrás, az NYTK-NerKor építésénél is. Az NYTK-NerKor korpuszt társszerzővel építettem, a korpusz-építési feladatok közül a hozzájárulásom elsősorban a morfológiai elemzést tartalmazó annotációs réteg megtervezése és elkészítése volt. Később az annotációt a teljes korpuszra kiterjesztettem. A teljes KorKor beépült az NYTK-NerKor anyagába, valamint a KorKor építésekor kidolgozott munkafolyamat, eszközök és útmutatók lehetővé tették az NYTK-NerKor morfológiai egyértelműsítést

tartalmazó annotációjának az elkészítését. Az NYTK-NerKor jelenleg a legnagyobb gold standard minőségű korpusz a névelemannotációt és a morfológiai egyértelműsítést tekintve, ezekkel az elemzési rétegekkel pedig a koreferenciafeloldás területén is hasznos erőforrás lehet, hiszen a névelemek lehetséges résztvevői a koreferenciakapcsolatoknak.

A korpuszok mellett további adatbázisok is születtek, amelyek a többértelmű névmási anafora jelenségére épültek. Ezek az adatbázisok mind olyan szövegeket tartalmaznak, amelyekben megjelenik egy kétértelmű névmási anafora, amelynek feloldása az ember számára triviális feladat, ám számítógépes nyelvészeti szempontból kifejezetten nehéz kihívás. Az adatbázisok mind már létező, más nyelvű adatok fordításával készültek, a munkát társszerzővel végeztem.

A dolgozat második felében anafora- és koreferenciafeloldáshoz, illetve a témához szorosan kapcsolódó zérónévmás-beszúráshoz kapcsolódó eszközöket ismertettem.

Szabályalapú megoldást kínáltam a zérónévmások beszúrájának feladatára, amelyet a KorKor építésekor előannotálásra használtam, így az annotátoroknak csak ellenőrizniük kellett a beillesztett névmásokat. A szabályalapú zérónévmás-beszúró emZero néven bekerült az `emt_sv` moduljai közé, így könnyen használható a szövegfeldolgozó keretrendszeren belül a zérónévmások beszúráására. A huBERT modell finomhangolásával egy tudásszegény, neurális megoldást is kínáltam a zérónévmás-beillesztés feladatára, amihez a KorKor korpuszt használtam tanító- és tesztadatként.

Implementáltam egy szabályalapú anaforafeloldó algoritmust, amelyet a KorKor építésekor előannotálásra használtam. Az annotátoroknak ellenőrizniük kellett a szabályalapú anaforafeloldó kimenetét, ám mivel az anaforafeloldó megoldásom csak a jelenségek egy szűk körét célozták, az anaforikus kapcsolatok több fajtáját, valamint a koreferenciakapcsolatokat is kézzel kellett beilleszteniük a szövegekbe.

Ismertettem egy társszerzővel készített neurális koreferenciafeloldót, amelyet a huBERT finomhangolásával készítettünk el. A finomhangoláshoz a KorKor és a SzegedKoref korpuszok anyagát összevontan használtuk, az annotációjukat egységesítettük. A megfelelő eredmény érdekében szófajalapú előszűrést is alkalmaztunk.

A disszertáció zárásaként bemutattam egy kísérletet, amelyben a ChatGPT-t teszteltem a magyar többértelmű névmások antecedenskeresésének feladatán. A kísérletekhez a disszertációban bemutatott HAPP adatbázis anyagát használtam. Igyekeztem többféle kérdésfeltevési módszerrel, több ismétléssel megvizsgálni, hogy hogyan teljesít az alkalmazás a többértelmű névmási anaforák feloldásának feladatán, mennyire konzisztens válaszokat ad, illetve milyen magyarázatokkal látja el a válaszokat.

A dolgozatban ismertetett erőforrásokat és eszközöket a reprodukálhatóság jegyében készítettem. Igyekeztem hangsúlyt fektetni a dokumentációra, hogy az általam készített korpuszok és programok más projektek számára is hasznosíthatók legyenek. Az annotációs útmutatók mellett, hogy lehetővé teszik a korpuszok bővítését a későbbiekben, a korpuszok annotációjának megértését is segítik a korpusz felhasználói számára. Szintén az elérhetőséget és az újrafelhasználhatóságot segíti, hogy az erőforrások szabadon hozzáférhető szövegek felhasználásával készültek, az eszközök pedig szintén nyílt licenc alatt érhetőek el. A doktori munka során az eszközök és erőforrások elkészítésekor több szemszögből jártam körül a számítógépes anafora- és koreferenciafeloldás témáját, valamint az ezekhez kapcsolódó egyéb alfeladatokat.

12. fejezet

Summary

The thesis covered the topic of computer-based anaphora and coreference resolution, approached from two perspectives: from the perspective of resources and tools. In the first half of the thesis, I described some resources that focus on these linguistic phenomena. I was able to use the presented resources as training and test data during my doctoral research.

The KorKor coreference corpus received several linguistic annotation layers, including anaphora and coreference annotation. All annotation layers are of manually checked quality, so in addition to anaphora and coreference relations, the corpus also includes gold standard quality morphological disambiguation and syntactic analysis, as well as manually inserted zero verbs, ellipitd verbs and zero pronouns. Although its annotation scheme differs from that of SzegedKoref, which can be considered a precedent, the two corpora can be used together as training and test data. KorKor successfully entered the CorefUD collection, which collects and harmonizes the coreference corpora of different languages, which increased the visibility of the resource. During the construction of KorKor, some additional useful tools were created, such as emDiff, which can be used to compare texts, prepare evaluations and calculate interannotator agreement between annotators, as well as converters between Hungarian morphological tagsets. These tools can be used as modules in the `emt sv` framework.

The material of the corpus and the processes and tools of construction proved to be useful in the construction of another resource, NYTK-NerKor. I built the NYTK-NerKor corpus with a co-author, among the corpus building tasks, my contribution was primarily the designing and preparation of the morphological annotation layer. Later, I extended the annotation to the entire corpus. The entire KorKor was integrated into the material of NYTK-NerKor, and the workflow, tools and guidelines

developed during the construction of KorKor made it possible to prepare the annotation of NYTK-NerKor with morphological disambiguation. NYTK-NerKor is currently the largest gold standard quality corpus in terms of name annotation and morphological disambiguation, and with these analysis layers it can also be a useful resource in the field of coreference resolution, since named entities are possible participants in coreference relationships.

In addition to the corpora, other databases were created, which were based on the phenomenon of ambiguous pronominal anaphora. These databases all contain texts in which an ambiguous pronominal anaphora appears, the resolution of which is a trivial task for humans, but from the point of view of computational linguistics, it turns out to be a particularly difficult task. The databases were all created by translating existing data in other languages, I did the work with a co-author.

In the second half of the thesis, I described tools related to anaphora and coreference resolution, as well as zero pronoun insertion, which is closely related to the topic.

I offered a rule-based solution to the task of inserting zero pronouns, which I used for pre-annotation when building KorKor, so the annotators only had to check the inserted pronouns. The rule-based solution for inserting zero pronouns named *emZero* has been added to the modules of *emtsv*, so it can be easily used to insert zero pronouns within this text processing framework. By fine-tuning the huBERT model, I also offered a neural solution to the task of zero pronoun insertion, for which I used the KorKor corpus as training and test data.

I implemented a rule-based anaphora resolution algorithm, which was used for pre-annotation in KorKor. The annotators had to check the output of the rule-based anaphora solver, but since my anaphora solver only targeted a narrow range of phenomena, they had to manually insert several types of anaphora as well as coreference relationships into the texts.

I introduced a neural coreference resolver created with a co-author, which we created by fine-tuning huBERT. For fine-tuning, we combined the data of the KorKor and SzegedKoref corpora, and standardized their annotations. In order to obtain a suitable result, we also applied pre-filtering based on part of speech tags.

At the end of the dissertation, I presented an experiment in which I tested ChatGPT on the task of searching for antecedents of Hungarian ambiguous pronouns. I used the data from the HAPP database presented in the dissertation for the experiments. I tried to use several methods of asking questions, with several repetitions, to examine how the application performs the task of resolving ambiguous pronoun anaphoras, how consistent the answers are, and what explanations it provides for the answers.

I created the resources and tools described in the thesis in the spirit of reproducibility. I tried to emphasize the documentation, so that the corpora and tools I created could also be used for other projects. In addition to enabling the expansion of corpora in the future, the annotation guidelines also help users to understand the annotation of the corpora. Accessibility and reusability are also facilitated by the fact that the resources were created using freely available texts, and the tools are also available under an open license. During the doctoral work, when preparing the tools and resources, I looked at the topic of anaphora and coreference resolution and other related subtasks from several perspectives.

13. fejezet

Felhasznált irodalom

- Amsili Pascal és Seminck Olga. A Google-Proof Collection of French Winograd Schemas. In: *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, Valencia, Spain. Association for Computational Linguistics, April, 2017, 24–29.
- Artstein Ron és Poesio Massimo. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 2008, 34(4): 555–596. Extended version: <https://cswww.essex.ac.uk/Research/nle/arrau/icagr.pdf>.
- Baptista J., Pereira Simone és Mamede Nuno J. ZAC : Zero Anaphora Corpus A Corpus for Zero Anaphora Resolution in Portuguese. In: *Proceedings of Workshop on Corpora and Tools for Processing Corpora, PROPOR 2016*, 2016.
- Barrault Loïc, Bojar Ondřej, Costa-jussà Marta R., Federmann Christian, Fishel Mark, Graham Yvette, Haddow Barry, Huck Matthias, Koehn Philipp, Malmasi Shervin, Monz Christof, Müller Mathias, Pal Santanu, Post Matt és Zampieri Marcos. Findings of the 2019 Conference on Machine Translation (WMT19). In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, Florence, Italy. Association for Computational Linguistics, 2019, 1–61.
- Bernard Timothée és Han Ting. Mandarinograd: A Chinese Collection of Winograd Schemas. In: *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, May, 2020, 21–26.
- Bobrow Daniel G. Natural Language Input for a Computer Problem Solving System. Technical report, USA, 1964.
- Brennan Susan E., Friedman Marilyn W. és Pollard Carl J. A Centering Approach to Pronouns. In: *Proceedings of the 25th Meeting of the Association for Computational Linguistics*, 1987, 155–162.
- Copeland B. J. The Turing Test*. *Minds and Machines*, 2000, 10:519–539.
- Csendes D., Csirik J. és Gyimóthy Tibor. The Szeged Corpus: A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus. In: Sojka Petr, Kopeček Ivan és Pala Karel szerk. *Text, Speech and Dialogue Lecture Notes in Computer Science*, 3206. kötet. Springer Berlin / Heidelberg, 2004, 41–47.
- Csendes Dóra, Csirik János, Gyimóthy Tibor és Kocsor András. The Szeged Treebank. In: *Proceedings of the 8th International Conference, TSD 2005*, Karlovy Vary, Czech Republic. Springer, September, 2005, 123–131.
- Damassino Nicola. The Questioning Turing Test. *Minds and Machines*, 12, 2020, 30:1–25.
- Davis Ernest, Morgenstern Leora és Ortiz Charles L. The First Winograd Schema Challenge at IJCAI-16. *AI Magazine*, Oct., 2017, 38(3):97–98.
- de Melo Gabriela S., Imaizumi Vinicius A. és Cozman Fabio G. Winograd Schemas in Portuguese. In: *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, Porto Alegre, RS, Brasil. SBC, 2019, 787–798.
- Doddington George, Mitchell Alexis, Przybocki Mark, Ramshaw Lance, Strassel Stephanie és Weischedel Ralph. The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA), May, 2004.

- É. Kiss Katalin. *The Syntax of Hungarian*. Cambridge University Press, Cambridge,, Egyesült Királyság, 2004.
- É. Kiss Katalin, Kiefer Ferenc és Siptár Péter. *Új magyar nyelvtan*. Osiris Kiadó, Budapest, 2003.
- Eckart de Castilho Richard, Mújdricza-Maydt Éva, Yimam Seid Muhie, Hartmann Silvana, Gurevych Iryna, Frank Anette és Biemann Chris. 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)*, Osaka, Japan. The COLING 2016 Organizing Committee, December, 2016, 76–84.
- Emelin Denis és Sennrich Rico. Wino-X: Multilingual Winograd Schemas for Commonsense Reasoning and Coreference Resolution. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, November, 2021, 8517–8532.
- Erjavec Tomaž. *MULTEXT-East Morphosyntactic Specifications. Version 3.0*, May, 2004.
- Frawley William. *Linguistic Semantics*. L. Erlbaum Associates, 1992.
- Frazier Lyn és Fodor Janet Dean. The Sausage Machine: A New Two-Stage Parsing Model. *Cognition*, 1978, 6(4):291–325.
- Ghaddar Abbas és Langlais Phillippe. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In: Chair Nicoletta Calzolari (Conference, Choukri Khalid, Declerck Thierry, Goggi Sara, Grobelnik Marko, Maegaard Bente, Mariani Joseph, Mazo Helene, Moreno Asuncion, Odijk Jan és Piperidis Stelios szerk. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA), may, 2016.
- Ghazal Ahmad, Ivanov Todor, Kostamaa Pekka, Crolotte Alain, Voong Ryan, Al-Kateb Mohammed, Ghazal Waleed és Zicari Roberto V. BigBench V2: The New and Improved BigBench. In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, 2017, 1225–1236.
- Guillou Liane, Hardmeier Christian, Smith Aaron, Tiedemann Jörg és Webber Bonnie. ParCor 1.0: A Parallel Pronoun-Coreference Corpus to Support Statistical MT. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA), May, 2014, 3191–3198.
- Hajič Jan, Ciaramita Massimiliano, Johansson Richard, Kawahara Daisuke, Martí Maria Antònia, Màrquez Lluís, Meyers Adam, Nivre Joakim, Padó Sebastian, Štěpánek Jan, Straňák Pavel, Surdeanu Mihai, Xue Nianwen és Zhang Yi. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Association for Computational Linguistics, 2009, 1–18.
- Hamp Birgit és Feldweg Helmut. GermaNet - a Lexical-Semantic Net for German. In: *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997.
- Hendrickx Iris és Hoste Veronique. Coreference Resolution on Blogs and Commented News. In: *Proceedings of the 7th Discourse Anaphora and Anaphor Resolution Colloquium on Anaphora Processing and Applications, DAARC '09*, Berlin, Heidelberg. Springer-Verlag, 2009, 43–53.
- Hinrichs Erhard, Kübler Sandra, Naumann Karin, Telljohann Heike és Trushkina Julia. Recent developments in linguistic annotations of the TüBa-D/Z treebank. In: Kübler Sandra, Nivre Joakim, Hinrichs Erhard és Wunsch Holger szerk. *Proceedings of the Third Workshop on Treebanks and Linguistic Theories*, Tübingen. 2004.
- Hirschman Lynette és Chinchor Nancy. Appendix F: MUC-7 Coreference Task Definition (version 3.0). In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998.

- Hirst Graeme. Anaphora in Natural Language Understanding: A Survey. In: *Lecture Notes in Computer Science*, 1981.
- Hobbs Jerry R. Resolving pronoun references. *Lingua*, 1978, 44(4):311–338.
- Honnibal Matthew. Introducing spaCy, 2015. <https://explosion.ai/blog/introducing-spacy>.
- Hu Junjie, Ruder Sebastian, Siddhant Aditya, Neubig Graham, Firat Orhan és Johnson Melvin. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization, 2020.
- Iida Ryu, Komachi Mamoru, Inui Kentaro és Matsumoto Yuji. Annotating a Japanese Text Corpus with Predicate-Argument and Coreference Relations. In: *Proceedings of the Linguistic Annotation Workshop*, Prague, Czech Republic. Association for Computational Linguistics, June, 2007, 132–139.
- Indig Balázs, Sass Bálint, Simon Eszter, Mittelholcz Iván, Vadász Noémi és Makrai Márton. One format to rule them all – The `emtsv` pipeline for Hungarian. In: *Proceedings of the 13th Linguistic Annotation Workshop*, Florence, Italy. Association for Computational Linguistics, 2019a, 155–165.
- Indig Balázs, Sass Bálint, Simon Eszter, Mittelholcz Iván, Kundráth Péter és Vadász Noémi. emtsv – Egy formátum mind felett. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XV. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. 2019b, 235–247.
- Indig Balázs és Vadász Noémi. Window in Human Parsing – How Far can a Preverb Go? In: Tadić Marko és Bekavac Božo szerk. *Proceedings of the Tenth International Conference on Natural Language Processing (HrTAL2016)*, Cham. Springer, 2016. to appear.
- Indig Balázs, Vadász Noémi és Kalivoda Ágnes. Decreasing Entropy: How Wide to Open the Window? In: Martín-Vide Carlos, Mizuki Takaaki és Vega-Rodríguez Miguel A. szerk. *Proceedings of Theory and Practice of Natural Computing: 5th International Conference (TPNC 2016)*, Cham. Springer, 2016, 137–148.
- Joshi Mandar, Levy Omer, Zettlemoyer Luke és Weld Daniel. BERT for Coreference Resolution: Baselines and Analysis. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, November, 2019, 5803–5808.
- K. Molnár Emese és Dömötör Andrea. Gondolatok a gondola-tokról, morfológiai annotációt javító módszerek tesztelése gold standard korpuszon. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XIX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2022)*, Szeged. Szegedi Tudományegyetem Informatikai Intézet, 2023, 341–353.
- Kiefer Ferenc szerk. *Strukturális magyar nyelvtan, Mondattan*. Akadémiai Kiadó, Budapest, 1992.
- Kocijana Vid, Davis Ernest, Lukaszewicz Thomas, Marcuse Gary és Morgenstern Leora. The Defeat of the Winograd Schema Challenge, 2023.
- Komlósy András. *Régensek és vonzatok*, 279–529. Akadémiai Kiadó, Budapest, 1992.
- Kovács Viktória. *Névmási anaforafeloldási kísérletek a magyar nyelvben*. Doktori disszertáció, Szegedi Tudományegyetem, 2021.
- Landis J. Richard és Koch Gary G. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 1977, 33(1): 159–174.
- Lapshinova-Koltunski Ekaterina, Ferreira Pedro Augusto, Lartaud Elina és Hardmeier Christian. ParCorFull2.0: A parallel corpus annotated with full coreference. In: *Proceedings of the 13th Conference on Linguistic Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2022, 805–813.

- Lee Kenton, He Luheng és Zettlemoyer Luke. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, June, 2018, 687–692.
- Lejtovicz Katalin és Kardkovács Zsolt Tivadar. Anaphora Resolution. In: *Proceedings of the 8th International Symposium of Hungarian Researchers on Computational Intelligence*, Budapest. 2007.
- Levesque Hector J., Davis Ernest és Morgenstern Leora. The Winograd Schema Challenge. In: *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2012, 552–561.
- Ligeti-Nagy Noémi, Ferenczi Gergő, Héja Enikő, Jelencsik-Mátyus Kinga, Laki László János, Vadász Noémi, Yang Zijian Győző és Váradi Tamás. HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából. In: *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. Szegedi Tudományegyetem, Informatikai Intézet, 2022, 431–446.
- Miháltz Márton. Tudásalapú koreferencia- és birtokviszony-feloldás magyar szövegekben. *Általános Nyelvészeti Tanulmányok*, 2012, 24:151–166.
- Miháltz Márton, Hatvani Csaba, Kuti Judit, Szarvas György, Csirik János, Prószéky Gábor és Váradi Tamás. Methods and Results of the Hungarian WordNet Project. In: *Proceedings of The Fourth Global WordNet Conference*, 2008, 311–321.
- Miháltz Márton, Naszodi Mátyás, Vajda Péter és Varasdi Károly. NP-koreferenciák feloldása magyar szövegekben a Magyar WordNet ontológia segítségével. In: Tanács Attila és Csendes Dóra szerk. *V. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016)*, Szeged. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 2007, 138–146.
- Mitkov Ruslan. Anaphora Resolution: The State Of The Art. Technical report, 1999.
- Mittelholcz Iván. emToken: Unicode-képes tokenizáló magyar nyelvre. In: Vincze Veronika szerk. *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017)*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2017, 70–78.
- Moosavi Nafise Sadat és Strube Michael. Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-based Entity Aware Metric. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics, August, 2016, 632–642.
- Morgenstern Leora, Davis Ernest és Ortiz Charles L. Planning, Executing, and Evaluating the Winograd Schema Challenge. *AI Magazine*, Apr., 2016, 37(1):50–54.
- Munkácsy Gergely és Farkas Richárd. Statisztikai koreferenciafeloldó rendszer magyar nyelvre — első eredmények. In: Tanács Attila, Varga Viktor és Vincze Veronika szerk. *XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016)*, Szeged. Szegedi Tudományegyetem, Informatikai Intézet, 2016, 295–297.
- Nedoluzhko Anna, Novák Michal, Popel Martin, Žabokrtský Zdeněk, Zeldes Amir és Zeman Daniel. CorefUD 1.0: Coreference Meets Universal Dependencies. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, June, 2022, 4859–4872.
- Nemeskey Dávid Márk. *Natural Language Processing methods for Language Modeling*. Doktori disszertáció, Eötvös Loránd University, 2020.
- Nemeskey Dávid Márk. Introducing huBERT. In: Berend Gábor szerk. *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021)*, Szeged. 2021, 3–14.
- Neufeld Eric és Finnstad Sonje. In defense of the Turing test. *AI & SOCIETY*, 12, 2020, 35.

- Ng Vincent. Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb., 2017, (1).
- Novák Michal, Popel Martin, Žabokrtský Zdeněk, Zeman Daniel, Nedoluzhko Anna, Acar Kutay, Bourgonje Peter, Cinková Silvie, Cebiroğlu Eryiğit Gülşen, Hajič Jan, Hardmeier Christian, Haug Dag, Jørgensen Tollef, Kåsen Andre, Krielke Pauline, Landragin Frédéric, Lapshinova-Koltunski Ekaterina, Mæhlum Petter, Martí M.Antònia, Mikulová Marie, Nøklestad Anders, Ogrodniczuk Maciej, Øvrelid Lilja, Pamay Arslan Tuğba, Recasens Marta, Solberg Per Erik, Stede Manfred, Straka Milan, Toldova Svetlana, Vadász Noémi, Velldal Erik, Vincze Veronika, Zeldes Amir és Žitkus Voldemaras. Coreference in Universal Dependencies 1.1 (CorefUD 1.1), 2022. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Novák Attila. Milyen a jó Humor? In: *I. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. SZTE, 2003, 138–144.
- Novák Attila. A New Form of Humor – Mapping Constraint-Based Computational Morphologies to a Finite-State Representation. In: Calzolari Nicoletta, Choukri Khalid, Declerck Thierry, Loftsson Hrafn, Maegaard Bente, Mariani Joseph, Moreno Asuncion, Odijk Jan és Piperidis Stelios szerk. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA), may, 2014.
- Novák Attila, Rebrus Péter és Ludányi Zsófia. Az emMorph morfológiai elemző annotációs formalizmusa. In: *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2017)*, Szeged. 2017, 70–78.
- Novák Attila, Siklósi Borbála és Oravecz Csaba. A New Integrated Open-source Morphological Analyzer for Hungarian. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), may, 2016a.
- Novák Attila, Siklósi Borbála és Oravecz Csaba. A New Integrated Open-source Morphological Analyzer for Hungarian. In: Calzolari Nicoletta, Choukri Khalid, Declerck Thierry, Goggi Sara, Grobelnik Marko, Maegaard Bente, Mariani Joseph, Mazo Helene, Moreno Asuncion, Odijk Jan és Piperidis Stelios szerk. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA), may, 2016b.
- Ogrodniczuk Maciej, Głowińska Katarzyna, Kopeć Mateusz, Savary Agata és Zawislawska Magdalena. Polish Coreference Corpus. In: Vetulani Zygmunt, Uszkoreit Hans és Kubis Marek szerk. *Human Language Technology. Challenges for Computer Science and Linguistics*, Cham. Springer International Publishing, 2016, 215–226.
- Orosz György és Novák Attila. PurePos 2.0 – an open source morphological disambiguator. In: Sharp B. és Zock M. szerk. *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*, Wroclaw. 2012, 53–63.
- Orosz György és Novák Attila. PurePos 2.0: a hybrid tool for morphological disambiguation. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA, September, 2013, 539–545.
- Pléh Csaba. *A mondatmegértés a magyar nyelvben*. Osiris tankönyvek. Osiris, 1998.
- Pléh Csaba és Radics Katalin. „Hiányos mondat”, pronominalizáció és a szöveg. *Általános Nyelvészeti Tanulmányok*, 1976, 11(1): 261–277.
- Popel Martin, Žabokrtský Zdeněk és Vojtek Martin. Udapi: Universal API for Universal Dependencies. In: *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, Gothenburg, Sweden. Association for Computational Linguistics, May, 2017, 96–101.

- Pradhan Sameer, Marcus Mitch, Palmer Martha, Ramshaw Lance és Weischedel Ralph. OntoNotes: A Unified Relational Semantic Representation. 09, 2007a, 517–526.
- Pradhan Sameer, Moschitti Alessandro, Xue Nianwen, Uryupina Olga és Zhang Yuchen. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In: *Joint Conference on EMNLP and CoNLL - Shared Task*, Jeju Island, Korea. Association for Computational Linguistics, July, 2012, 1–40.
- Pradhan Sameer S., Ramshaw Lance, Weischedel Ralph, MacBride Jessica és Micciulla Linnea. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In: *Proceedings of the International Conference on Semantic Computing, ICSC '07*, Washington, DC, USA. IEEE Computer Society, 2007b, 446–453.
- Prószéky Gábor és Indig Balázs. Magyar szövegek pszicholingvisztikai indíttatású elemzése számítógéppel. *Alkalmazott nyelvtudomány*, 2015, 15(1-2):29–44.
- Prószéky Gábor, Indig Balázs és Vadász Noémi. Performanciaalapú elemző magyar szövegek számítógépes megértéséhez. In: Bence Kas szerk. *”Szavad ne feledd!”: Tanulmányok Bánréti Zoltán tiszteletére*. MTA Nyelvtudományi Intézet, Budapest, 2016, 223–232.
- Rahman Altaf és Ng Vincent. Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, 777–789.
- Recasens Marta és Martí Antonia. AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language Resources and Evaluation*, 12, 2010, 44:315–345.
- Reimers Nils és Gurevych Iryna. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11, 2019.
- Roesiger Ina és Kuhn Jonas. IMS HotCoref DE: A Data-driven Co-reference Resolver for German. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia. European Language Resources Association (ELRA), May, 2016, 155–160.
- Sakaguchi Keisuke, Bras Ronan Le, Bhagavatula Chandra és Choi Yejin. WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale. *CoRR*, 2019, abs/1907.10641.
- Sass Bálint. Élő vagy élettelen? In: Tanács Attila és Csendes Dóra szerk. *V. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. Szegedi Tudományegyetem, 2007, 195–203.
- Schroff Florian, Kalenichenko Dmitry és Philbin James. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR*, 2015, abs/1503.03832.
- Shavrina Tatiana, Fenogenova Alena, Emelyanov Anton, Shevelev Denis, Artemova Ekaterina, Malykh Valentin, Mikhailov Vladislav, Tikhonova Maria, Chertok Andrey és Evlampiev Andrey. RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark. *arXiv preprint arXiv:2010.15925*, 2020.
- Simon Eszter, Indig Balázs, Kalivoda Ágnes, Mittelholcz Iván, Sass Bálint és Vadász Noémi. Újabb fejlemények az e-magyar háza táján. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020)*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2020, 29–42.
- Simon Eszter és Nemeskey Dávid Márk. Automatically generated NE tagged corpora for English and Hungarian. In: *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, Jeju, Korea. Association for Computational Linguistics, July, 2012, 38–46.

- Simon Eszter és Vadász Noémi. Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus. In: Ekstein Kamil, Pártl František és Konopík Miloslav szerk. *Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, ProceedingsLecture Notes in Computer Science*, 12848. kötet. Springer, 2021, 222–234.
- Steinberger Ralf, Ebrahim Mohamed, Poulis Alexandros, Carrasco-Benitez Manuel, Schlüter Patrick, Przybyszewski Marek és Gilbro Signe. An overview of the European Union’s highly multilingual parallel corpora file. *Language Resources and Evaluation*, 2014, (48):679–707.
- Steinberger Ralf, Pouliquen Bruno, Widiger Anna, Ignat Camelia, Erjavec Tomaž, Tufis Dan és Varga Dániel. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC’2006)*, Genoa, Italy. 2006.
- Stenetorp Pontus, Pyysalo Sampo, Topić Goran, Ohta Tomoko, Ananiadou Sophia és Tsujii Jun’ichi. 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, EACL ’12*, Stroudsburg, PA, USA. Association for Computational Linguistics, 2012, 102–107.
- Sun Yu, Wang Shuohuan, Feng Shikun, Ding Siyu, Pang Chao, Shang Junyuan, Liu Jiaxiang, Chen Xuyi, Zhao Yanbin, Lu Yuxiang, Liu Weixin, Wu Zhihua, Gong Weibao, Liang Jianzhong, Shang Zhizhou, Sun Peng, Liu Wei, Ouyang Xuan, Yu Dianhai, Tian Hao, Wu Hua és Wang Haifeng. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *ArXiv*, 2021, abs/2107.02137.
- Szarvas György, Farkas Richárd, Felföldi László, Kocsor András és Csirik János. A highly accurate Named Entity corpus for Hungarian. In: *Electronic Proceedings of the 5th International Conference on Language Resources and Evaluation*, May, 2006.
- Tiedemann Jörg. Parallel Data, Tools and Interfaces in OPUS. In: Chair) Nicoletta Calzolari (Conference, Choukri Khalid, Declerck Thierry, Dogan Mehmet Ugur, Maegaard Bente, Mariani Joseph, Odijk Jan és Piperidis Stelios szerk. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*, Istanbul, Turkey. European Language Resources Association (ELRA), 2012.
- Tihanyi László és Merényi Csaba. A MetaMorpho fordítóprogram projekt 2006-ban. In: Alexin Zoltán és Csendes Dóra szerk. *IV. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. Szegedi Tudományegyetem, 2006, 169–179.
- Tikhonov Alexey és Ryabinin Max. It’s All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning, 2021.
- Tjong Kim Sang Erik F. és De Meulder Fien. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Daelemans Walter és Osborne Miles szerk. *Proceedings of CoNLL-2003*. Edmonton, Canada, 2003.
- Turing A. Computing Machinery and Intelligence. *Mind*, 1950, 59(236):433–460.
- Vadász Noémi. Building a Manually Annotated Hungarian Coreference Corpus: Workflow and Tools. In: *Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*, Gyeongju, Republic of Korea. Association for Computational Linguistics, October, 2022, 38–47.
- Vadász Noémi. Resolving Hungarian Anaphora with ChatGPT. In: Ekstein Kamil, Pártl František és Konopík Miloslav szerk. *Text, Speech, and Dialogue*, Cham. Springer Nature Switzerland, 2023, 45–57.
- Vadász Noémi. Anaforafeloldás menet közben – névmások egy pszicholingvisztikailag motivált elemzőben. In: Ludányi Zsófia szerk. *Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből 2017: XI. Alkalmazott Nyelvészeti Doktoranduszkonferencia*. MTA Nyelvtudományi Intézet, Budapest, 2017, 192–205.

- Vadász Noémi. KorKorpusz: kézzel annotált, többretegű pilotkorpusz építése. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XVI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2020)*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2020, 141–154.
- Vadász Noémi és Indig Balázs. A birtokos esete az ablakkal. In: Scheibl György szerk. *LingDok: Nyelvész-doktoranduszok dolgozatai 17*. SZTE Nyelvtudományi Doktori Iskola, Szeged, 2018, 85–99.
- Vadász Noémi, Kalivoda Ágnes és Indig Balázs. Ablak által világosan – Vonzatkeret-egyértelműsítés az igekötők és az infinitívuszi vonzatok segítségével. In: Vincze Veronika szerk. *XIII. Magyar Számítógépes Nyelvészeti Konferencia*. Szegedi Tudományegyetem Informatikai Intézet, Szeged, 2017, 3–12.
- Vadász Noémi és Ligeti-Nagy Noémi. Winograd schemata and other datasets for anaphora resolution in Hungarian. *Acta Linguistica Academica*, 2022.
- Vadász Noémi és Nyéki Bence. Koreferenciafeloldás magyar szövegeken BERT-tel. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XIX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2022)*, Szeged. Szegedi Tudományegyetem Informatikai Intézet, 2023, 119–131.
- Vadász Noémi és Simon Eszter. Konverterek magyar morfológiai címkekészletek között. In: Berend Gábor, Gosztolya Gábor és Vincze Veronika szerk. *XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019)*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2019, 99–112.
- van Deemter Kees és Kibble Rodger. On Coreferring: Coreference in MUC and Related Annotation Schemes. *Computational Linguistics*, 2000, 26(4):629–637.
- Várádi Tamás. The Hungarian National Corpus. In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas de Gran Canaria. European Language Resources Association, 2002, 385–389.
- Vincze Veronika, Farkas Richárd, Simkó Katalin Ilona, Szántó Zsolt és Varga Viktor. Univerzális dependencia és morfológia magyar nyelvre. In: Tanács Attila, Viktor Varga és Veronika Vincze szerk. *XII. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2016, 322–329.
- Vincze Veronika, Hegedűs Klára és Farkas Richárd. SzegedKoref: kézzel annotált magyar nyelvű koreferenciakorpusz. In: Tanács Attila, Varga Viktor és Vincze Veronika szerk. *XI. Magyar Számítógépes Nyelvészeti Konferencia*. SZTE TTIK Informatikai Tanszékcsoport, 2015, 312–322.
- Vincze Veronika, Hegedűs Klára, Sliz-Nagy Alex és Farkas Richárd. SzegedKoref: A Hungarian Coreference Corpus. In: *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association, May, 2018.
- Vincze Veronika, Szauter Dóra, Almási Attila, Móra György, Alexin Zoltán és Csirik János. Hungarian Dependency Treebank. In: Chair Nicoletta Calzolari (Conference, Choukri Khalid, Maegaard Bente, Mariani Joseph, Odijk Jan, Piperidis Stelios, Rosner Mike és Tapias Daniel szerk. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA), 2010.
- Vincze Veronika, Varga Viktor, Simkó Katalin Ilona, Zsibrita János, Nagy Ágoston és Farkas Richárd. Morfológiai újítások a Szeged Korpusz 2.5-ben. In: Tanács Attila, Viktor Varga és Veronika Vincze szerk. *X. Magyar Számítógépes Nyelvészeti Konferencia*, Szeged. Szegedi Tudományegyetem Informatikai Tanszékcsoport, 2014, 332–338.
- Wang Alex, Pruksachatkun Yada, Nangia Nikita, Singh Amanpreet, Michael Julian, Hill Felix, Levy Omer és Bowman Samuel R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems, 2020a.

- Wang Alex, Singh Amanpreet, Michael Julian, Hill Felix, Levy Omer és Bowman Samuel. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium. Association for Computational Linguistics, November, 2018, 353–355.
- Wang Bochao, Xu Hang, Zhang Jiajin, Chen Chen, Fang Xiaozhi, Xu Yixing, Kang Ning, Hong Lanqing, Jiang Chenhan, Cai Xinyue, Li Jiawei, Zhou Fengwei, Li Yong, Liu Zhicheng, Chen Xinghao, Han Kai, Shu Han, Song Dehua, Wang Yunhe, Zhang Wei, Xu Chunjing, Li Zhenguo, Liu Wenzhi és Zhang Tong. VEGA: Towards an End-to-End Configurable AutoML Pipeline. *ArXiv*, 2020b, 2011.01507.
- Webster Kellie, Recasens Marta, Axelrod Vera és Baldrige Jason. Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. *Transactions of the Association for Computational Linguistics*, 2018, 6:605–617.
- Weischedel Ralph, Palmer Martha, Marcus Mitchell, Hovy Eduard, Pradhan Sameer, Ramshaw Lance, Xue Nianwen, Taylor Ann, Kaufman Jeff, Franchini Michelle, El-Bachouti Mohammed, Belvin Robert és Houston Ann. OntoNotes Release 5.0, 2013. <https://hdl.handle.net/11272.1/AB2/MKJJ2R>.
- Weiss Dawid. Korpus Rzeczpospolitej, 2002. <http://www.cs.put.poznan.pl/dweiss/rzeczpospolita>.
- Wilks Yorick. *An Intelligent Analyzer and Understander of English*, 61–82. Springer Netherlands, Dordrecht, 2007.
- Winograd Terry. Understanding natural language. *Cognitive Psychology*, 1972, 3(1):1–191.
- Wolf Thomas. State-of-the-art neural coreference resolution for chatbots, 2017. <https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30>.
- Zsibrita János, Vincze Veronika és Farkas Richárd. magyarlan: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: *Proceedings of RANLP*, 2013, 763–771.