DOI:10.15774/PPKE.ITK.2018.007

Egy sok szálon futó nyelvelemző program moduljainak kialakítása és harmonizációja

Indig Balázs

PhD disszertáció

Témavezető:
Dr. Prószéky Gábor
az MTA doktora

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar
Roska Tamás Műszaki és Természettudományi Doktori Iskola

Budapest, 2017

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

Összefoglaló

A nyelvtechnológiában a szerelőszalag (pipeline) egy régóta ismert és alkalmazott soros architektúra. Napjaink bő kínálata a felhasználható, szabadon elérhető eszközökben megteremti azt az igényt, hogy az egyes folyamatok együttműködve és akár párhuzamosan funkcionáljanak. A dolgozatban körüljárom a megoldott feladatok felépítését, és definiálom a még megoldatlan feladatokkal kapcsolatos problémákat, melyeket részben önmagukban is megpróbálok orvosolni.

A szekvenciális címkézéssel megoldható feladatcsalád tagjai közötti hasonlóságok és különbségek feltárásával jobban megérthetővé válik, hogy egyes módszerek miért működnek és mások miért nem. A részletes elemzés után javasolt megoldásaim a magyar főnévi csoportok és az angol közvetlen összetevős szerkezetek keresésének feladatában meghaladják a state-of-the-art módszereket. Az utóbbi feladaton demonstrálom továbbá, hogy a különféle szekvenciális címkézési eljárások hogyan javíthatóak általánosságban a hagyományosan használt címkekészleteknek a korpusz alapján történő módosításával (lexikalizáció). Továbbá bemutatok egy módszert, amivel a korpuszból nyert paraméterek transzformálhatók a korpuszok között.

Megteremtem a magyar igei argumentumokat leíró vonzatkeret-leírások szemantikai információinak feldúsításával a pontosabb osztályozás lehetőségét azáltal, hogy az angol és magyar nyelvű erőforrásokat összekapcsolom. Végül pedig ismertetem az AnaGramma elemző rendszer eredendően párhuzamos architektúráját, amely a bemutatott eredményeket és tapasztalatokat is felhasználva jött létre, valamint néhány eddig megoldatlan nyelvi jelenség kezelését.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

Tartalomjegyzék

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

1. fejezet

Bevezetés

„A trivialitásnak fokozatai is vannak...”
(Szalay Mihály, Lineáris algebra gyak.
2005, ELTE IK)

1.1. A nyelvtechnológiai szerelőszalag

Hagyományosan a nyelvtechnológiai eszközök egy *csővezeték (pipeline)* formájában (hívják még *szerelőszalagnak* is) működnek az architektúra minden előnyével és hátrányával. A csővezeték a nyers szövegtől indul és az elemzés tetszőleges szintjén ér véget. A tradicionális modulok a következők:

- Mondatra bontó, tokenizáló
- Morfológiai elemző, szófaji egyértelműsítő
- Főnévicsoport- és névelemkereső (sekély elemzés)
- Szintaktikai elemző
- Szemantikai elemző
- Információkinyerő, gépi fordító, stb.

A csővezeték egyszerűségéből adódóan az egyes moduloknak elméletben nem kell tudniuk semmit a többi modulról. Így hagyományosan a tesztelésük is a *gold*

DOI:10.15774/PPKE.ITK.2018.007

sztenderd korpuszon vagy más néven referenciaadaton (gold standard) történt, azaz a tökéletes bemenetből tökéletes kimenetet kellett előállítaniuk. Viszont az éles használat során a csővezetéknek a bemenettel nem közvetlenül érintkező moduljai az őket megelőző modulok hibáit felhalmozva, korántsem tökéletes bemenettel kell, hogy dolgozzanak. A szakirodalomban jellemzően kevés utalást találunk arra, hogy mennyire robusztusak ezek a rendszerek egy csővezeték részeként, hibás bemenet esetén[1].

Ez a tény motiválja azt a kérdést, hogy hogyan működhetnének jobban együtt ezek a modulok, hogy a potenciális hibák ne halmozódjanak fel a feldolgozás során a csővezetékben. Ezért dolgozatomban áttekintem a szabadon elérhető, magyar nyelvű state-of-the-art módszereket, valamint megoldást keresek a harmonizációjukkal kapcsolatos problémákra.

1.2. A korpuszok

A dolgozatban használt, különböző típusú gépi tanulást végző programok és mintakereső módszerek a szabványos körülmények közötti összehasonlítás és elemzés nélkül nem tudnak tudományosan értékelhető eredményt adni. Ezért a használt programok kiértékeléséhez a tudományterületen megszokott, a *pontosság* és *fedés* harmonikus közepeként előálló *F-mértéket*, bemenetként és elvárt kimenetként pedig a főbb elérhető korpuszokat használtam, melyeket a következő bekezdésben részletesen bemutatok.

A tudományos vizsgálatokra szánt szövegek speciális szempontok szerint előállított korpuszok formájában érhetőek el. A korpuszok tartalmazzák a szövegekhez tartozó, többségében automatikusan készült annotációt. A felügyelt tanításra épülő módszereknek szüksége van referenciaadatra is, mely egy előre meghatározott formátum és eljárásrend szerint kézzel készül. Az ilyen korpuszok előállítási költsége az annotáció emberierőforrás-igénye miatt igen magas. Magyar nyelvre

[1]Itt érdemes megjegyezni, hogy a fősodratú kutatások a modulok együttműködésének vizsgálata helyett azok egybeolvasztását vizsgálják, így a karakter és bájt alapú nyelvfeldolgozást (Costa-jussà, Escolano és Fonollosa 2017; Mogren és Johansson 2017), mely egy teljesen más megoldási stratégia a felvetett problémára. A módszer általánosságából adódóan hatékony, ámbár eredmény-centrikusan eltávolodik a vizsgált folyamatok megértésétől, így ezt a módszercsaládot a dolgozatban a továbbiakban nem tárgyalom.

DOI:10.15774/PPKE.ITK.2018.007

a *Szeged Korpusz* (Csendes, Hatvani et al. 2003) a jelenleg egyetlen kézzel annotált korpusz, mely 70 000 mondatot és 1 194 348 tokent tartalmaz. A függőségi elemzéssel ellátott változata a *Szeged Treebank* (Vincze et al. 2010). A dolgozatban magyar nyelvre ezt a korpuszt használtam én is tanítóanyagként a maximális főnévi csoportok kereséséhez.

A nyelv modellezéséhez viszont elegendő a lehető legnagyobb mennyiségű szöveg gyűjtése, mivel a feldolgozás emberi erőforrást nem igényel. Ezekkel a szövegekkel kapcsolatos egyedüli kritérium, hogy a megfelelő nyelven legyenek és normalizált formában, egységes egészt alkossanak. Az internetes kommunikáció erősödésével manapság nagyon könnyű különböző minőségű szövegeket szisztematikusan legyűjteni az internetről[1], így a magyar nyelvre elérhető, géppel elemzett korpuszok száma is egyre nő.

A dolgozatban nyelvmodellezésre két korpuszt használtam. Az egyik a *Magyar Nemzeti Szövegtár* első és második (2.0.3) verziója (Oravecz, Váradi és Sass 2014), az első változat 187 millió, a második pedig 785 millió szót (978 millió tokent) tartalmaz változatos forrásokból (beszélt szövegek átiratai, határon túli újságok, jogi szövegek, parlamenti naplók, stb.). A második korpusz pedig a teljes egészében az internetről gyűjtött szövegekből készült *Pázmány Korpusz*, mely 1,2 milliárd szavas (Endrédy 2016).

A dolgozatban szerepel továbbá az *InfoRádió Korpusz*, amely csak szerkesztett rövidhíreket tartalmaz, néha többmondatos megnyilatkozások formájában. 2 millió szavával egy kisebb doménspecifikus korpuszt képez, mely az 1.5.1. fejezetben bemutatott elemzőmodell által feldolgozni kívánt szövegek prototípusát alkotja. Az angol nyelvű *közvetlen összetevők keresését* célzó vizsgálatokat pedig a *CoNLL-2000 korpuszon* (Tjong Kim Sang és Buchholz 2000) (259 104 token) végeztem.

[1] Az állítás alátámasztására egy technikai jellegű vizsgálatot végeztem a témában (Indig 2018).

DOI:10.15774/PPKE.ITK.2018.007

1.3. A már megoldott feladatok: a szavak szintje

1.3.1. Tokenizálás és mondatra bontás

A helyesen írt magyar nyelvű szövegek[1] mondatokra és tokenekre bontása egyszerű feladat. A mondatok többnyire előre meghatározott írásjellel végződnek és nagybetűvel kezdődnek. A szavak tokenekre bontásánál a magyar tipográfiától eltérő szövegek kezelésére is fel kell készülnünk. Az angoltól eltérő idéző- és gondolatjel formátum miatt a felhasználók nem mindig veszik a fáradságot, hogy tipográfiailag megfelelő szöveget szerkesszenek. A lehetőségek még így is elég korlátozottak. Az idéző- és zárójelek a szavak egyik végén szerepelnek, a bal oldalukon nyitó-, míg a jobb oldalukon záróelemként funkcionálnak. Hibalehetőség lehet az, hogy a nem mondatvéget jelző pontokat is leválasztjuk a szavakról, így az olyan felismert entitások, mint a dátumok, rövidítések, római számok és sorszámnevek védettséget kapnak. Ezek jól leírhatók reguláris kifejezésekkel, valamint megadhatók rövidítéslistákkal.

A fentiek figyelembevételével nyilvánvalóan adódik, hogy elsősorban szabályalapú megoldások születtek a magyar tokenizálásra. Az első ilyen a *Huntoken* (Miháczi, Németh és Rácz 2003) nevű program volt, ami a *GNU Flex* lexikális elemző generátor saját leírónyelvén[2] íródott, és egymás utáni reguláris kifejezésekből álló szabályokat tartalmaz több külön fájlban, melyek egy csővezetékben sorban egymás után hívódnak meg. Az egyes modulok nagyban építenek a szöveg lokális sajátosságaira és az egyszerű szűrők egymás utáni futtatására. A programot a *Szeged Korpuszon* tesztelték 98%-os eredménnyel.

A Huntoken nagy hátránya az, hogy nem visszaállítható tokenizálást végez, valamint a korszerű Unicode karaktereket nem képes kezelni a Flex motor miatt. Több kísérlet volt a Huntoken alapjain nyugvó új implementáció elkészítésére, mely a kor követelményeinek megfelel. Ilyen volt a *PureToken*[3] (Indig 2013) és az azon szerzett tapasztalatokon alapuló, az *e-magyar* rendszeren belül működő

[1] A roncsolódott, rosszul formázott szövegeken érdemes először normalizáló eljárásokat futtatni, mert azok célzottan tudják a szöveget javítani, aminek köszönhetően nem lesz szükségtelenül nagy a tokenizálás és mondatra bontás komplexitása.

[2] `https://github.com/westes/flex`

[3] A saját technikai kontribúcióm.

DOI:10.15774/PPKE.ITK.2018.007

emToken[1] (Mittelholcz 2017). Az *emToken*, az eredeti specifikációja szerint a *Huntoken* kimenetével egyező kimenetet állít elő, így ez az eszköz sem képes detokenizálható kimenetet létrehozni, de a későbbi fejlesztések nyomán ez a funkció implementálásra került.

1.3.2. Számítógépes morfológia

A magyar nyelv az agglutináló nyelvek osztályába tartozik. Az angolhoz képest sokkal gazdagabb a morfológiai eszközkészlete. Emiatt a tanítóanyagból hiányzó, úgynevezett *Out of Vocabulary (OOV)* szavak aránya sokkal magasabb, mint az angolban. Szükséges tehát külön modult építeni a probléma kezelésére, mely modellezni tudja a természetes nyelv morfológiájának működését. A magyar nyelv morfológiája nyelvészeti szempontból jól kutatott témának számít, emiatt két független, szabályalapú gépi morfológia is létezik, habár napjainkig nem született kellően jó, széleskörűen használt statisztikai alapú gépi tanulással felépíthető morfológiai modell.

A *Hunmorph* (Trón et al. 2005) a több nyelven széleskörben használt Hunspell[2] módszerét vette alapul, azaz hogy a különböző osztályokba sorolt szavakhoz toldalékosztályokba sorolt folytatási szabályokkal modellezte a morfológia működését. A *Humor* (Novák 2003) a Morphologic Kft. fejlesztése. Nem nyílt forrású, és a teljesen saját kódrendszert, az úgynevezett *Humor kódot* használja. A belső motorja egy unifikáción alapuló nyelvtan, amelyben a különféle jegyek bitenként vannak felvéve és ezek a számítógép által gyorsan kezelhetőek. A fenti két gépi morfológia egyesítéseként jött létre a Helsinki Finite-State Transducer Technológiára (HFST) (Lindén et al. 2013) épülő *emMorph* (Novák, Rebrus és Ludányi 2017), mely szabadon elérhető kutatási célra.

Az említett eszközök nem képesek statisztikai információt, mint például valószínűséget, gyakoriságot vagy konfidenciaértéket rendelni a kimenetként adott lehetőségekhez, amelyet később a statisztikai programok fel tudnának használni, viszont jól tükrözik a magyar nyelv produkciós szabályait, paradigmáit amelyeket jelenleg statisztikai alapon nem lehet elég jól modellezni.

[1] Az *emToken* az *e-magyar* digitális nyelvfeldolgozó rendszer – melyben én is részt vettem – része (Váradi, Simon, Sass, Gerőcs, Mittelholcz, Novák, Indig et al. 2017).

[2] `http://hunspell.github.io/`

DOI:10.15774/PPKE.ITK.2018.007

1.3.3. Szófaji egyértelműsítés

A magyar nyelvű szófaji egyértelműsítésben az jelenti a kihívást, hogy a tanítóanyagban nem szereplő, OOV szavakat hogyan kezeljük. Ellentétben a csővezetékben előrébb található lépésekkel, itt felügyelt statisztikai módszereteket alkalmaznak. Azon belül is szinte változatlan formában a *T'n'T* (Brants 2000) rendszerből átalakított és végül *OCaml* nyelven újraimplementált *HunPOS* (Halácsy, Kornai és Oravecz 2007), valamint az abból *JAVA* nyelven újjászületett *PurePOS* (Novák, Orosz és Indig 2011; Orosz és Novák 2013) rendszereket érdemes megemlíteni[1].

A PurePOS az elődeitől abban különbözik, hogy nemcsak magyar nyelvre adaptált *végződésfelismerővel (suffix guesser)* rendelkezik, melyet a szerzője a HunPOS alapjaira implementált, hanem egyúttal a szavak lemmájának meghatározására is képes. Bár voltak egyéb próbálkozások, mint például a statisztikai gépi fordítás alapú HuLaPOS (Laki, Orosz és Novák 2013) és különböző címkézők kombinációja, nem értek el kellően jó eredményt, és így gyakorlati szempontból a PurePOS jobban használható.

A módszer működése főbb vonalakban a következő: az első lépésben, az *emissziós* vagy *unigram modellben* a *guesser modul* egy szóhoz felsorol több lehetséges kimenetet a hozzájuk tartozó valószínűségekkel ($\boldsymbol{Q(w_i|t_i)}$) egy adott valószínűségi eloszlás szerint. A második lépésben, a *címkeátmenet-modellben* egy címkesorozatokon tanított trigram-modell ($P(t_i|t_{i-1}, t_{i-2})$) a címkéket a Viterbi algoritmus, vagy annak egy megszorított változata (beam search) alapján (Forney 1973) megpróbálja a mondat összes címkéje sorrendjének figyelembevételével optimalizálni. Az eljárás a Markov tulajdonság[2] felhasználásával a klasszikus képlet szerint működik – melyet kis változtatásokkal más szekvenciális címkézési feladatokban is használnak:

$$\underset{t_1 \ldots t_T}{\operatorname{argmax}} \left[\prod_{i=1}^{T} P(t_i|t_{i-1}, t_{i-2}) \mathbf{Q(w_i|t_i)} \right] \mathbf{P(t_{T+1}|t_T)} \tag{1.1}$$

[1] A PurePOS fejlesztésének technikai oldalában részt vettem, illetve napjainkban én tartom karban a kódot.

[2] A Markov tulajdonság azt jelenti, hogy az adott elem osztálya csak véges darab közvetlen megelőző elemtől függ, ami alapján a mondatszinten legvalószínűbb átmenetsorozat kiszámolható a Viterbi algoritmussal.

DOI:10.15774/PPKE.ITK.2018.007

A képletben színessel jelölt részek külön magyarázatot érdemelnek, mivel eltérnek a klasszikus *Hidden Markov Modelltől (HMM)*: a modern implementációkban a kék színnel jelölt Q valószínűségi függvény, mely az emissziós modellt adja, támaszkodhat a szó és a címke együttes előfordulásának feltételes valószínűségén kívül jellemzőkre is[1]. A pirossal jelölt rész pedig a T'n'T címkézőben megjelent újítás, mely a mondat végén elhelyezett extremális elem felhasználásával pontosabb eredményt ad a mondatvégi tokenek esetén[2].

Érdemes megjegyezni, hogy bár statisztikai módszerről van szó, a felügyelt tanításhoz szükséges kézzel annotált tanítóanyag is, valamint a state-of-the-art eredményt csak úgy éri el a szófaji egyértelműsítő program, ha egy szabályalapú morfológia által támogatottan hozza meg a döntéseit az egyes szavak töveinek tekintetében. Vegyük észre tehát, hogy bár a három modul (kézzel annotált korpusz, szabályalapú morfológia és a statisztikai elven működő szófaji egyértelműsítő) függetlennek látszik, a legjobb eredmény elérése érdekében egy modulként szükséges működniük. Minden formai, szabványbeli vagy elvi eltérés nagyban ront a teljesítményen.

Az utóbbira jó példa – bár távolabbról kapcsolódik, de ide tartozik a tokenizáló kérdése –, hogy külön tokenként kezeljük-e az *-e* partikulát vagy nem[3]. Mivel a rendszerben használt tokenizáló szabályalapú, ezért szükséges, hogy a többi szabályalapú rendszerrel egyeztetve működjön, mert erre „automatikusan" csak a felügyelet nélküli rendszerek esetén lenne lehetőség, ami csak azt követelné meg, hogy a rendszer tanítására használt és így az alapját képező korpusz egy és ugyanaz legyen minden modul számára[4].

Bár megoldott feladatnak számít a szófaji egyértelműsítés a kutatás szempontjából, a technikai háttér még fejlesztésre szorul. Az egyes almodulok mélyebb együttműködése, azaz hogy hogyan lehetne a magyar morfológia működését sta-

[1] Amennyiben az unigram modell jellemzőket használ, általában a maximum entrópia módszert alkalmazzák a feltételes valószínűség helyett.

[2] Ezt az eljárást átvezettem az általam készített HunTag3 programba, melyet a 2.4.1. fejezetben ismertetek.

[3] Az *-e* partikula a szófaji egyértelműsítő szempontjából jó, ha külön van, mert akkor a végződésmodellnek nem kell külön megtanulni a hozzá kapcsolódó szavak *-e* végződését.

[4] A felügyelet nélküli tanulási módszer viszont várhatóan a teljesítmény romlásával járna.

DOI:10.15774/PPKE.ITK.2018.007

tisztikai módszerekkel kellően jól modellezni, úgy hogy az legalább kiegészítse, ha nem is lekörözze a szabályalapú módszert, még megoldatlan kutatási kérdés.

Általánosságban láthatjuk, hogy a hibrid rendszerek működésének alapvető eszközei a kézzel annotált korpusz, a szabályalapú rendszer és a statisztikai alapú rendszer, melyek a legalapvetőbb tervezési lépésektől az együttműködés céljával jöttek létre. Mit sem ér egy olyan szabályalapú formalizmus, amely nem igazolható vissza statisztikai eszközökkel – legyen az bármilyen tetszetős elméleti szempontból –, ugyanis a statisztikai módszerek másként nem tudnak együttműködni vele és ilyenformán nem lehet egy hibrid rendszer része. A csővezeték szempontjából pedig az is fontos, hogy a moduljai robusztusak legyenek, hogy a hibák felerősítését csökkenteni tudják.

1.4. Az általam vizsgált feladatok: frázisok és megnyilatkozások

A magyar nyelvben az egyszerű mondatok két jól elkülöníthető komponensre bonthatók. Az egyik a *közvetlen összetevős szerkezetek*, ezekben az elemek sorrendje kötött, és nem mozognak szabadon a mondatban. Ilyenek a *főnévi csoportok*. Míg a másik osztály az *igei szerkezetek*, melynek elemei között megtalálhatjuk az imént említett közvetlen összetevős szerkezeteket mint az igék argumentumait. A következő fejezetekben ezt a két osztályt fogom bővebben tárgyalni.

1.4.1. Közvetlen összetevők megtalálása a mondatban

A főnévi csoportok mint a közvetlen összetevős szerkezetek legprominensebb fajtája a magyarban azért különösen érdekesek, mert kötött nyelvtanuknak (Kornai 1985; Recski 2014) köszönhetően jól azonosíthatóak. Ez az állítás viszont csak akkor állja meg a helyét, amennyiben minimális főnévi csoportról beszélünk.

A továbbiakban a megjelölt csoportokat címkézett zárójelekkel jelölöm, mert a jelölés előnye a szemléletessége mellett az, hogy könnyen alakítható át ez egyes szavakhoz rendelt zárójelállapot- és csoportnév-kombinációkká, melyből a gépek számára könnyen feldolgozható *szekvenciális címkézési feladatot* lehet csinálni. A következő fejezetekben nyelvészeti szempontból ismertetem a főnévi csoportok

DOI:10.15774/PPKE.ITK.2018.007

két nyelvtechnológiai aspektusból fontos osztályát, gépi kezelésük problémáit és felismerésük technikáját pedig a 2. fejezetben mutatom be.

1.4.2. A minimális főnévi csoportok tulajdonságai

A *minimális főnévi csoport (minNP, bázisNP)* egyik definíciója szerint olyan NP, ami önmagában már nem tartalmaz NP-t (Ramshaw és Marcus 1995).

A gyakorlatban az NP-k fő elemei (determináns, jelző, főnév) elhagyhatóak, mivel az elhagyott elem referál – többnyire a kontextusból származó – már ismertnek tekintett szereplőre vagy máshonnan „kiszámolható”[1]. Az NP utolsó, nyelvtani esetet hordozó eleme a csoport függőségi értelemben vett feje. Szórendjét tekintve a főnévi csoport végén nem csak főnév állhat, hanem zárhatja melléknév, melléknévi igenév, névmás és névutó is.

Felszíni szempontból nem tudunk foglalkozni azzal az esettel, ha nem csak részek, hanem a teljes szerkezet is elhagyható, mivel ilyenkor más elemekből kell „kiszámolni” az elhagyott elemet. Az esetek többségében viszont legalább egy elem jelzi a főnévi csoport jelenlétét: az a mondatban jelenlevő bizonyos elem viszont szinte bármelyik lehet abban az esetben, ha az elhagyott részektől eltekintve az amúgy rendkívül kötött sorrend helyes. Az alábbi négy példából látható a minNP néhány különböző esete.

Az (1) példában egy módosítóval[2] bővített NP látható. Szintaktikai szempontból az olvasónak jelzés, hogy ha talál egy módosító, akkor tőle balra kell keresnie az opcionális determinánst, illetve jobbra az elhagyható főnevet.

(1) *A cirmos cica elment aludni .*
DET MN.NOM FN.NOM IGE.ME3 IGE.INF PUNCT
[A cirmos cica]$_{\text{NP}}$ elment aludni .

[1] Például a tulajdonnevek és a birtokolt főnévi csoportok mindig determináltak ezért a determináns ilyenkor többnyire nem szerepel a mondatban.

[2] A főnevet több különböző szófajú szó (melléknév, melléknévi igenév, számnév, bizonyos ragozott névutók, stb.) módosíthatja, melyekre a dolgozatban egységesen módosítóként fogok hivatkozni.

DOI:10.15774/PPKE.ITK.2018.007

A (2) példában a módosító nélküli NP látható. A nem jelen levő módosító arra utal, hogy vagy ismert vagy a megnyilatkozás szempontjából irreleváns tulajdonságokkal bír az NP feje.

(2) *A cica elment aludni .*
DET FN.NOM IGE.ME3 IGE.INF PUNCT
[A cica]$_{\text{NP}}$ elment aludni .

A (3) példában a determinálatlan NP látható. A mondat felszólító módban van. Ez szükségessé teszi, hogy ismert legyen az NP által megjelölt szereplő, ami így determinált, tehát a determináns elhagyható.

(3) *Gyere ki , cirmos cica !*
IGE.PE2 IK PUNCT MN.NOM FN.NOM PUNCT
Gyere ki , [cirmos cica]$_{\text{NP}}$!

A (4) példában a „fej nélküli” NP látható. Függőségi szempontból mindig kell a szerkezetnek egy elem, ami a feje lehet, de jelen példában a fejként funkcionáló elem – vélhetően a kontextus miatt – elhagyásra került, ezért az azt megelőző elem, az (utolsó) főnevet módosító token kapta meg a nominatívuszi esetragot és így a fej szerepét. Koreferenciális szempontból egy üres elem van jelen a szerkezetben, mely összeköthető egy ismert szereplővel, akit a módosító alapján azonosít a beszélő.

(4) *A legkisebb mindig éhezett .*
DET MN.SUPL.NOM HAT IGE.ME3 PUNCT
[A legkisebb]$_{\text{NP}}$ mindig éhezett .

A feldolgozás szempontjából a minimális főnévi csoportok csak egy alsóbbrendű lépést jelentenek, mivel általában a minimális főnévi csoportokból akár többszöri bővítéssel létrejött szerkezetek az igék argumentumai, így a szereplők[1] is. Ezért figyelmünket a következő fejezetekben a főnévi csoportok sorozataira és a nagyobb egységeikre, a *maximális főnévi csoportokra* irányítjuk, melyek még tartogatnak tudományos kihívást a gépi feldolgozásban.

[1]Neo-Davidsoniánus értelemben (Hobbs 1985) nem csak az igék argumentumai és azok részei, hanem maguk az igék is mint események.

DOI:10.15774/PPKE.ITK.2018.007

1.4.3. A maximális főnévi csoportok tulajdonságai

Maximális főnévi csoport (maxNP) definíció szerint azon szerkezet, mely bővítés nélkül egy, egy vagy többszöri bővítéssel több minimális főnévi csoportból áll elő (Váradi 2003) jellemzően a következő módokon (melyeket az (5) példa illusztrál):

- minimális főnévi csoport, amely nincs bővítve (lásd az (1, 2, 3, 4) példák),
- két (nem csak minimális) főnévi csoport összekapcsolva konjunkcióval (lásd az (5a) példa),
- két (nem csak minimális) főnévi csoport összekapcsolva participiummal (lásd az (5b) példa),
- két (nem csak minimális) főnévi csoport összekapcsolva birtokos szerkezettel (lásd az (5c) példa),
- két (nem csak minimális) főnévi csoport összekapcsolva konkatenációval (Ligeti-Nagy 2016) (lásd az (5d).

(5) a. *A legkisebb és legnagyobb testvér jóban volt .*
DET MN.SUPL.NOM KOT MN.SUPL.NOM FN.NOM HAT IGE.ME3 PUNCT
[A legkisebb és legnagyobb testvér]$_{\text{NP}}$ jóban volt .

b. *A szószt magára kenő fiú volt a legkisebb .*
DET FN.ACC FN.NM.SUB IGE.OKEP.NOM FN.NOM IGE.ME3 DET MN.SUPL.NOM PUNCT
[A szószt magára kenő fiú]$_{\text{NP}}$ volt [a legkisebb]$_{\text{NP}}$.

c. *A fiú nagyobbik testvére mindig kedves volt .*
DET FN.NOM MN.NOM FN.PSe3.NOM HAT MN.NOM IGE.Me3 PUNCT
[A fiú nagyobbik testvére]$_{\text{NP}}$ mindig kedves volt .

d. *Angela Merkel német kancellár felszólalt .*
FN.NOM FN.NOM MN.NOM FN.NOM IGE.Me3 PUNCT
[Angela Merkel német kancellár]$_{\text{NP}}$ felszólalt .

DOI:10.15774/PPKE.ITK.2018.007

A fenti példák csak egy kis szeletét mutatják a konstrukciók nyújtotta nyelvi lehetőségeknek. Azért is fontos külön megemlíteni őket, mert ezek a maximális főnévi csoportok lesznek az igék argumentumai a szintaxis és a szemantika szintjén is. Sikeres gépi feldolgozásuk épp ezért nagyon fontos. A számítógépes kezelésükkel kapcsolatos problémákat és a problémák egy részére adott megoldásaimat a 2.1. fejezetben részletesen tárgyalom.

1.4.4. Igei szerkezetek

Ha az ige a mondat fejem akkor szemantikai értelemben az ige maga a predikátum és az argumentumai a predikátum argumentumai. Ezért nagyon fontos szerepet tölt be a mondatelemzés során. Az igei szerkezet az adott ige tövéből és az annak vonzataiból alkotott *vonzatkeretből* áll. Mivel az igék argumentumai a mondatban szinte tetszőleges sorrendben szerepelhetnek, az egyes vonzatkeretek nem tesznek különbséget az argumentumok sorrendjében[1].

Jogosan merül fel a kérdés, hogy hogyan jutunk hozzá ezekhez a vonzatkeretekhez. Két megoldás született erre a problémára. Az egyik a szakértők által kézzel alkotott adatbázis, mely a MetaMorpho szabályalapú fordítórendszer alapjául szolgált (Prószéky, Tihanyi és Ugray 2004). A másik pedig a felügyelt gépi tanulásból származó szintaktikai elemző által megelemzett korpuszok szintaxisfáinak felhasználásával előállított Mazsola (Sass 2009) adatbázis[2]. Mindkét módszernek megvan a maga előnye és hátránya. A szabályalapú rendszer egyenrangúként kezeli a szabályokat – melyek fedése rendkívül nagy – függetlenül a gyakoriságuktól (akárcsak az említett szabályalapú morfológia) és egy rendkívül komplex keretrendszerbe ágyazva tárolja őket, mely nem lett statisztikai alapon ellenőrizve, így az együttműködése a statisztikai rendszerekkel kérdéses. A Mazsola rendszer viszont túlzott leegyszerűsítéseket alkalmaz és csak a legegyszerűbb, legnyilvánvalóbb szerkezeteket tartalmazza, mivel a célja sokkal inkább a pontosság és nem a fedés. Az igei vonzatkeret-adatbázisokat és a velük kapcsolatos kutatásomat bővebben a 4.1. fejezetben tárgyalom.

[1] A valóban előforduló sorrendek meghatározására egy mondatvázakat leíró kutatás kezdődött (Endrédy 2014).

[2] Itt azzal az egyszerűsítéssel élek, hogy a többi, a Mazsolához hasonló, statisztikai alapon felépülő rendszert nem említem, csak a 4.1. fejezetben mutatom be őket.

DOI:10.15774/PPKE.ITK.2018.007

1.4.5. Szintaktikai és szemantikai elemzés

A *Syntactic Structures* (Chomsky 1957) óta a nyelvészek és a számítógépes nyelvészek külön próbálják kezelni a szintaxist a szemantikától. Meghonosodott az az elmélet, hogy a szintaktikai elemzőnek egy egyértelmű mondatfát kell rendelnie a mondathoz, és ezt adjuk át a szemantikai elemzőnek. Viszont sok esetben a szemantika nélkül nem lehet egyértelműsíteni magát a mondatfát sem. Elég itt egy egyszerű példára gondolni: „Lelőttem egy elefántot a pizsamámban." (J. Fodor és Lepore 2004) Anélkül a szemantikai tudás nélkül, hogy a pizsama szempontjából a két szereplő nem felcserélhető (bár szintaktikailag azok), nem dönthető el, hogy a két elemzés közül melyik a helyes. Ennek ellenére, az emberi elemző számára a mondat egyértelmű, és nem okoz nehézséget a megértése. Ez a szigorúan generatív elmélet, bár eredeti formájában a nyelvészetben manapság sok kritika[1] éri[2] (Domaradzki 2007), mégis a ma is használt formalizmusokon felfedezhető az öröksége.

A szintaktikai elemzésben két alapvető formalizmus van használatban. A **közvetlen összetevős elemzés** – amikor a közvetlen összetevőkre redukált mondatcsonkok egy hierarchiába épülnek fel – a nyelvtechnológia hajnalán azért alakult ki, mert a teljes elemzés elég erőforrás-igényes és rossz minőségű volt. Az akkoriban egyeduralkodónak számító egyszerűbb feladatokat hatékonyan lehetett megoldani a közvetlen összetevős elemzéssel. Ezekben a feladatokban a mondatoknak csak bizonyos részeire volt szükség, például információkinyeréshez vagy keresési szavak, tárgymutatók készítéséhez. Manapság is legtöbbször csak az első szintet építik meg a fában (lásd az 1.4.1. fejezetben ismertetett feladatot), mivel a többire nincs szükség, vagy pedig az összetartozó szerkezetek nem közvetlenül jönnek egymás után és más módszerrel kell folytatni az elemzést.

A közvetlen összetevős elemzés lényege, hogy az összetevők megtalálása után egy külön fázisban az egymás mellett lévő összetevőket addig vonják össze nagyobb összetevőkké, amíg egy összetevő nem marad, ami a mondatszimbólummal lesz egyenértékű. Így állnak elő a mondatfa különböző szintjei. A módszer nagy

[1] `https://blogs.scientificamerican.com/cross-check/is-chomskys-theory-of-language-wrong-pinker-weighs-in-on-debate/`

[2] Az emberi nyelveknél jóval egyszerűbb programozási nyelvek viszont még mindig ezen az elven működnek.

DOI:10.15774/PPKE.ITK.2018.007

hátránya, hogy feltételezi az egymás mellett fix sorrendben következő szavakból összeálló összetevők kizárólagos jelenlétét. A szabadabb szórendű nyelvek esetén, amelyekben az összetevők részei messze is kerülhetnek egymástól, a módszer nehezen alkalmazható. Mára az angol és a főbb nyelvek esetében teljesen marginalizálódott a szerepe a jó minőségű és gyors szintaktikai elemzők előretörésével, de ezek hiányában a magyarnál és a kisebb intenzitással kutatott nyelveknél még mindig szükséges lépés, és fejlesztés alatt állnak a módszerek.

A másik formalizmus a **függőségi elemzés**, mely az egyes szavak között függőségi relációkat feltételez, amivel egyértelmű alá-fölérendeltségi viszony hozható létre az egyes összetevők között. Az adott szerkezet legfőbb eleme a fej, mely alárendeltje lehet egy nagyobb szerkezetnek egy további függőségi viszonyon keresztül. A módszer orvosolja a szórendfüggőség problémáját, hiszen nincs megkötés a szavak mondatbeli helyére. Hátránya viszont, hogy bárha géppel nagyon hatékony elemzők hozhatók is létre ebben a formalizmusban – egy valószínűségekre alapuló keresési teret elképzelve –, mégsem tudnak számot adni az egymás mellé rendelt elemekről, melyek esetén a sorrend megváltoztatásával más szintaktikai reprezentációt kapunk, hiszen az egyik elemet a másik alá kell rendelni valamilyen rögzített szabály szerint.

A fent említett módszerek közös hiányossága, hogy nem adnak számot arról, ami az „emberi elemzőben” történik, mivel ez nem is céljuk. Ezen felül továbbra is küzdenek a szükségtelen többértelműség problémájával, ami az emberi elemzőnek nem okoz gondot.

1.5. Motiváció: nem jó, hogy az eszközök elszigetelten működnek

Dolgozatomban azt vizsgálom, hogy miként lehetne a legjobb módszereket úgy tovább javítani, hogy az együttműködésüknek köszönhetően jobban hasonlítsanak az emberi elemző működésére.

DOI:10.15774/PPKE.ITK.2018.007

1.5.1. Egy pszicholingvisztikailag reális elemzőmodell

A fent bemutatott, információelméleten alapuló elemzők előnye, hogy gyorsan működnek, mivel tervezésüknél fogva elsődleges szempont volt a hatékony matematikai apparátus használata – ami jóval meghaladja az emberi elemző kapacitását – , és valószínűségi átmenetekkel operálva egy jó matematikai tulajdonsággal rendelkező speciális gráfot, egy *mondatfát* hoznak létre. Közös hiányosságuk viszont, hogy az egyes modulok egymástól függetlenül próbálnak működni, ezért olyan hibákat vétenek, melyek az emberi elemző számára elképzelhetetlenek. Erre a problémára válaszként, a mesterséges neurális hálózatok előtérbe kerülésével egy időben a feladat tovább egyszerűsödött: több tudományterületen is egységesen az adott bemenetből előállítandó adott kimenet létrehozása lett az elsődleges cél a mögöttes matematikai modell egységesítésével, mely számtalan izgalmas új eredménnyel kecsegtet.

Habár az új megoldási stratégia előnye, hogy a korábbi széttagoltságból erdő hibákat megoldja, mégis eközben a folyamat mélyebb, a neurális hálózatok működési sajátosságaitól független feladatspecifikus részleteinek megértése teljes mértékben háttérbe szorul. Ebből következik, hogy az előbbinél a formalizmus tervezési hibáiból, az utóbbinál pedig a gépek sajátosságaiból fakadóan nem nyerünk betekintést az emberi elemzőben található rendszer működésébe. Különös tekintettel arra, hogy hogyan oldja fel azokat a többértelműségeket, amelyeket a gép szisztematikusan elront, valamint nem képes kezelni. Az emberi elemzés szem előtt tartása olyan korlátok bevezetését jelenti, amelyekkel a feladat láthatóan megoldható, de a jelenlegi számítógépes módszerek – legyenek azok hagyományosan sorosak vagy újabban párhuzamosak – ezen korlátok nélkül sem képesek maradéktalanul teljesíteni a feladatot[1].

A fenti megfontolásból adódóan kezdte meg az *MTA–PPKE Magyar Nyelvtechnológiai Kutatócsoport* – melynek jelenleg is a tagja vagyok – a működését, hogy létrehozzon egy olyan elemzőmodellt (AnaGramma), amely orvosolja a fent vázolt problémákat, valamint amely hatékonyabb működést fog elérni (Prószéky és Indig 2015a; Prószéky, Indig, Miháltz et al. 2014; Prószéky, Indig és

[1] Továbbá az olyan korlátozások elhagyását is ide értjük, ami az adott matematikai formalizmus kedvéért került csak be a rendszerbe, organikus motiváció nincs a megtartására.

DOI:10.15774/PPKE.ITK.2018.007

Vadász 2016). Kutatócsoportunk arra vállalkozott, hogy a kurrens pszicholingvisztikai kutatások (Pléh 2014) figyelembevételével egy számítógépes elemzőrendszert hozzon létre, amely visszajelzésként és egyfajta *deszkamodellként (proof of concept, pilot)* szolgál a pszicholingvista kutatók számára.

1.5.2. Az elemző célkitűzései

***Pszicholingvisztikai* indíttatású:** ismert tény, hogy a nyelvi szerkezetek elemzés közbeni kiválasztása közben hozott döntéseink felül tudják bírálni a lexikont (Prószéky 2000). Az elemzés előtt előre rögzített, lexikonokat és ismereteket összegző erőforrásokat használó szabályalapú és az egyes szerkezetek korábbi gyakoriságára építő valószínűségi elemzők kizárólag a „múltbéli" ismereteikre és statisztikákra támaszkodva tudják meghozni döntéseiket (Brants és Crocker 2000)[1].

Az általunk létrehozandó elemzőmodellben olyan megoldást szándékozunk adni, melyben a bemenet – néha szokatlan felépítésének – elemzése közben „nincs zavaró hatása" a hiányos statisztikai adatoknak és a sokszor tévútra vezető szabályoknak. Kiinduló hipotézisünk az, hogy a nyelvhasználó fejében két rendszer él: egy a tanult szerkezetekre építő és egy aktuális döntéseket hozó rendszer, ami az emberi elemzőhöz hasonlóan a valós idejű feldolgozást akkor is képes megvalósítani, ha a „megtanult" szerkezetek egymásnak ellentmondó (például egymáshoz nem illeszkedő jegyszerkezeteket tartalmazó) nyelvtani információkat hordoznak. A modellünk kidolgozása során, amennyire csak lehet, az információelméletből és programozási nyelvek feldolgozásából ismert irodalomban tárgyalt hagyományos módszerektől eltérő, és sokkal inkább az emberi feldolgozásra jellemző működést részesítjük előnyben, még ha az összehasonlítás végett a hagyományos modulokat használjuk is – nem horizontálisan, hanem vertikálisan – kezdetben.

***Performancia alapú*:** a nyelvészetben nagy hatású generatív modellek a gépi feldolgozás szempontjából nem hatékonyak, mivel a valóságban előforduló, szerkesztetlen szövegek elemzésére nem alkalmasak. Ennek egyik oka, hogy a mondatfa levezetésében használt, nem invertálható transzformációk – melyek a ge-

[1]Megjegyzendő, hogy a legújabb kutatások iránya megegyezik a miénkkel, azaz a bemenethez adaptált általános modellekkel (Farajian et al. 2017) mintegy szimulálják a lexikon felülbírálását, amely az emberi elemzőben is végbemegy.

DOI:10.15774/PPKE.ITK.2018.007

neratív nyelvészetet valamilyen formában végigkísérik (Chomsky 1957) – főképp a mondatok előállítására alkalmasak, viszont az elemzésükre kevésbé. Léteznek ugyan transzformációmentes generatív modellek is, a „hatékony elemezhetőség" még azokban sem elsődleges szempont. Ennek oka, hogy az összes generatív módszer közös tulajdonsága – még akkor is, ha annak célja a hatékony elemzés – a kompetencia preferálása a performanciával szemben: azaz a valóságban előforduló, rosszul formált zajos szövegek nem képezik a vizsgálatok tárgyát, holott az emberi elemző az ilyen jellegű szövegekkel is elboldogul.

A *performancia-alapúság* az elemzőmodellünk számára azt jelenti, hogy minden nyelvi megnyilatkozás feldolgozandó, ami „előfordul"; viszont ami elvben ugyan lehetne, de valójában nem fordul elő, az valamilyen értelemben kevésbé lényeges. A nyelvi minták feltárásában tehát nem helyezünk hangsúlyt az elméletileg létező, de a gyakorlatban meglehetősen ritka jelenségek kezelésére. Ugyanakkor bármilyen – rosszul formált, agrammatikus – szöveget igyekszünk nyelvi megnyilvánulásnak tekinteni és értelmezni még akkor is ha csak részszerkezet ismerhető fel.

Szigorúan *balról jobbra*, szavanként dolgozza fel a bemenetet: az emberi elemzőhöz hasonlóan (McConkie és Rayner 1976; Rayner 1998) időben szigorúan előre – a megnyilatkozással egy időben, minimális pufferrel és ebből adódó késleltetéssel – balról jobbra szavanként halad, és nem képes figyelembe venni az „olvasás/hallás szempontjából még el nem hangzott", az aktuális elemtől jobbra található, időben későbbi elemeket, viszont igyekszik minden olyan információt felhasználni, mely a megnyilatkozás értelmezéséhez szükséges. Például a döntéshez igénybe tudja venni a teljes elhangzott baloldali kontextust, és különösen az annak elemeiből épített nagyobb szerkezeteket. A hagyományos grammatikai értelemben szerkesztetlen, nem tökéletes szövegekkel hasonlóan jár el, mint a szerkesztett megnyilatkozásokkal, akárcsak az emberi elemző. Modellünknek nincs módja az elemzés bizonyos pontján a megnyilatkozás még el nem hangzott vagy le nem írt részét felhasználni, vagy arra hivatkozni. Legfeljebb késleltetheti a döntést, vagy feltételezésekkel élhet a leírtak alapján – mint a hagyományos nyelvmodellek –, a megnyilatkozás végéig. Amikor az elemzett szó kikerül a pufferből és a bal kontextus részévé válik, az elemző nem tér hozzá vissza, az elemzése

DOI:10.15774/PPKE.ITK.2018.007

végleges. A fentiekből nem következik, hogy az adott pillanatban legvalószínűbb elemzést nem kell felülbírálni, visszalépésre kényszerítve az elemzőt, ugyanis az újraelemzés az emberi elemzőnél is előforduló jelenség a *kerti ösvény (garden path) jelenség* miatt (lásd a (6) példát (Pléh 1999)).

(6) *A* ***tanárnőd*** *megszerette a diák.*

A mondat végén látható, hogy csak akkor értelmes a mondat, ha a *tanárnő* a tárgy.

A kerti ösvényeket általában a hétköznapi kommunikációban – amennyiben tudatában vagyunk – kerüljük a grice-i maximák (Grice és Harman 1975) betartásából adódóan, és inkább csak viccek, vagy szándékos félrevezetés alkalmával fordulnak elő[1]. Ezért az AnaGramma – hipotézisem szerint az emberi elemzőhöz hasonlóan – igyekszik elkerülni a szövegben történő „ugrálást”[2], és csak a legvégső esetben folyamodik újraelemzéshez – ha a beállításai ezt írják elő –, a jövőben viszont várhatóan képes lesz az újraolvasással felfedezhető kerti ösvények célzott felderítésére is.

Architektúrája eredendően *párhuzamos*: a ma ismert számítógépes mondatelemzők szinte kizárólag egyirányú feldolgozást végeznek, azaz nincs oda-vissza kapcsolat a különböző nyelvi szintek között[3]. Ez ugyanis a hibák felhalmozódásához vezet, amire általában egy egyszerű gyakoriságon alapuló szűrő a gyakorlati megoldás. A megvalósítandó elemzőmodell viszont párhuzamos szálakon többféle nyelvi elemzést indít, melyekkel egyidejűleg jelennek meg más, a feldolgozandó szöveghez kapcsolható jelentést és világismeretet kezelő szálak. Elemző algoritmusunk tehát egyfajta konszenzust keres a különböző megértési stratégiák és eljárások között (Pléh 1999). Amint tehát a humán információfeldolgozásban, a mi elemzőnkben is egyidejűleg és szorosan működnek együtt a

[1] Érdemes megjegyezni, hogy számtalanszor utólag, sokszori olvasásra derül csak fény bizonyos kerti ösvények létére, mely önmagában mutatja, hogy beállítódásából adódóan az emberi elemző sokáig nem tekinti lehetséges „ösvénynek” ezeket az elemzési utakat.

[2] A rendszer működésének alapjául szolgáló, szemmozgást vizsgáló kísérletek magyar nyelvre tudomásom szerint még nem készültek, így az elméletünk ezen részét utólag kell pszicholingvisztikai szempontból alátámasztani.

[3] Eltekintve a mesterséges neurális hálóktól, ahol nem különböztetünk meg ilyen szinteket.

DOI:10.15774/PPKE.ITK.2018.007

nyelvi elemzést és az értelmezést végző modulok[1]. Modellünk további jellemzője, hogy párhuzamos, és az egyes modulok szoros együttműködésben, kommunikálva, mintegy egymást javítva működnek és ezáltal lerövidítik az elemzés idejét. A hozzáadott szemantikai információknak köszönhetően pedig egyértelmű elemzést tud adni az elemző a betáplált világismeret alapján.

Az akár több mondatból álló *megnyilvánulást* tekintjük alapegységnek: a hagyományos elemzők elemzési egysége a mondat, ami emiatt tévesen külön választja a mondaton túli koreferenciaviszonyok feloldását a mondaton belüliektől[2], megnehezítve az egységes kezelésüket. Modellünkben a mondatnál nagyobb, sokkal természetesebb szerkesztési egységet tekintjük alapegységnek: az akár több mondatból álló megnyilatkozásokat. Így lehetőség van a mondaton belüli és mondatok közötti anaforikus viszonyok egységes kezelésére. Így a részszerkezetek teljes összekapcsolása nem feltétlen egyetlen mondaton belül valósul meg, és a mondatközi szerkezetek is kezelhetővé válnak. A létrejött referenciális elemek ugyanezen reprezentációban megjelenő, akár mondatokon átívelő éleket vezethetnek be a reprezentációnkba.

Eltérünk a klasszikus *mondatfa-reprezentációtól*: a tervezett reprezentációnk a már ismertetett függőségi elemzés formalizmusához áll legközelebb. Mi is az elemek közötti függőségi viszonyokat jelöljük, de úgy, hogy a formalizmus elsődlegesen a nyelvi jelenségekhez igazítódjon, és ne fordítva.

Megvizsgáltuk tehát a hagyományos, kompetenciaalapú világ különböző, létező, hatékony függőségi elemzőit is, amilyen például a MaltParser (Nivre 2006), a Stanford Parser (De Marneffe, MacCartney és Christopher D Manning 2006), vagy a véges állapotú függőségi elemző (Oflazer 2003). Ezek valóban a nyelvi egységek egymás közötti viszonyainak leírását célozzák meg. Mégis erősen a mondatalapú, mondatok szerint szeparált feldolgozáshoz kötődnek, ezért nem

[1] Az újabban divatos, neurális hálós megközelítéstől eltérően, mi fenntartjuk az egyes modulok elkülönítését, hogy a párhuzamos működés során az interakcióikat vizsgálhassuk.

[2] Szerkeszthetünk egy mondatot úgy, hogy több tagmondatból álljon, és úgy is, hogy több független egyszerű mondatból. A koreferenciaviszonyok mindkét esetben azonosan működnek a megnyilatkozások belül. Akár átnyúlnak a mondaton, akár nem.

DOI:10.15774/PPKE.ITK.2018.007

voltak közvetlenül használhatók. Magyar nyelvre egyébként történtek már korábban függőségi megközelítések, mind szabályalapúak, mint például a holland DLT rendszerhez készített nyelvtan (Prószéky, Koutny és Wacha 1989), mind adatorientáltak, mint a Szeged Treebank függőségifa-formátumú változata (Vincze et al. 2010). A nyelvi jelenségek leírásának fedése szempontjából az eddig készített legátfogóbb magyar mondatelemző, a *MetaMorpho* fordítórendszer (Prószéky, Tihanyi és Ugray 2004), mely bár nem függőségi leíráson alapul, a rendelkezésünkre áll. Az összes fenti elemző közös tulajdonsága, hogy egyikük sem kezeli megfelelően a többértelműségek feloldását, rossz a hibatűrésük.

Az elemzéseink egységes reprezentációjaként ezért egy sajátos gráfot választottunk, amely alkalmas a szavak közötti függőségi viszonyoknak és a mondatok egyes részeinek – vonatkozó névmások, visszautalások kezelése stb. – referenciális alapon történő összekötésére úgy, hogy a mondatközi kapcsolatok is kódolhatóak benne. Szándékunk szerint egy-egy konjunktív elem jelenléte vagy hiánya nem okozhatja az azonos tartalom felszíni különbségek miatti radikálisan különböző feldolgozását, pusztán a mondathatárok különbözősége miatt. Az így kapott különböző típusú „függőségi jellegű" éleket tartalmazó – és sok esetben csak az elemzés végén összefüggővé váló – gráf, mely az irányított körmentes gráfok osztályába tartozik, jobban megfelel a szándékainknak, mint a hagyományos mondatfa-reprezentáció.

1.5.3. A főnévi csoportok és az igei szerkezetek egymást segítik

A fent bemutatottak alapján úgy gondolom, hogy ahhoz, hogy a főnévi csoportokról, felépítésükről és egymáshoz való viszonyaikról – a gépi kezelésükhöz szükséges – tisztább képet kapjunk, meg kell vizsgálni az igék vonzatkereteinek mondaton belüli viszonyait is. Viszont az igék vonzatkereteinek tisztázásához szükségesnek látom a főnévi csoportok típusainak és mondatbeli rendjének részletes vizsgálatát.

Ebből következően a dolgozatban ismertetett kutatásaimat több irányból kezdtem meg, abban a reményben, hogy az egyes részekben külön-külön elért eredmények felhasználhatóak lesznek a másik kiválasztott területen is. Dolgozatomban

DOI:10.15774/PPKE.ITK.2018.007

először a magyar és angol főnévi csoportok szekvenciális címkézéssel[1] történő felismerésének javításában elért eredményeim ismertetésével foglalkozom. Majd a főnévi csoportoktól fokozatosan távolodva – és a problémát egy másik perspektívába helyezve –, az angol főnévi csoportok tekintetében vizsgált, de sokkal általánosabb érvényű eredményeimet tárgyalom. Az ezt követő, az erőforrások összekapcsolásáról szóló fejezetben ismertetett igei vonzatkeret-adatbázisok kapcsán felmerült szabványossági problémák egy jövőbeni megoldását jelenthetik a főnévi csoportok tisztázásában elért eredményeim. Végül a már ismertetett, pszicholingvisztikailag motivált elemzőmodellel kapcsolatos, a főnévi és az igei szerkezeteket egységes elemzési keretrendszerbe foglaló eredményeimet tárgyalom egy közös fejezetben.A disszertációt az alkalmazási lehetőségek és a további lehetséges kutatási irányok ismertetését tartalmazó fejezet zárja.

[1]A szekvenciális címkézés szigorúan balról-jobbra halad a szövegen, mint a fent vázolt AnaGramma elemzőrendszer. Így mindkét rendszer hasonlít az emberi elemző olvasás közbeni balról jobbra haladásához.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

2. fejezet

Főnévi csoportok automatikus meghatározása

Csónakos: Ez csak rom. Ezt már romnak építették.
Nemecsek: Hát ha már építették, miért nem építettek új várat? Száz év múlva magától rom lett volna belőle...

(Molnár Ferenc: A Pál utcai fiúk)

2.1. A főnévi csoportok gépi felismerésének problémái

Az 1.4.3. fejezetben említett maximális főnévi csoportokra vonatkozó példák csak egy kis szeletét mutatják a konstrukciók nyújtotta nyelvi lehetőségeknek. Az igeneveknek saját vonzataik és szabad határozóik vannak, melyek nem tekinthetők közvetlenül az NP részének, mivel az igenévhez kapcsolódnak, mint egy finit igei szerkezetben. A maximális NP-ket alkotó minimális NP-k elemeiről viszont könnyen eldönthető, hogy egy token a főnévi csoport részét képezi-e vagy nem, hiszen az egyértelmű szófaji címkét kell csak nézni, és az alapján az osztályozás triviálisan elvégezhető.

Elméletben a megengedhető konstrukcióik ismerete folytán a maximális NP-k is jól kezelhetőek, mégis bizonyos szempontból ki tudnak lépni a közvetlen összetevős szerkezetek osztályából akkor, ha a birtok a birtokos szerkezet datívusszal jelölt tagjához képest eltávolodik és egy vagy több szó bekerül a két tag közé.

DOI:10.15774/PPKE.ITK.2018.007

Ilyenkor nem teszünk különbséget, hogy nulla vagy több token került-e be a két tag közé: egységesen két külön NP-nek tekintjük őket, pedig szemantikai szinten a szerkezet ekvivalens az egy NP-nek tekintett, nominatívusszal jelölt birtokos szerkezettel. Ezen kívül a konjunkciót is könnyen összekeverhetjük a tagmondatok összekapcsolásával. A kontextus és a szemantikai reprezentáció ismerete nélkül még az emberi elemzőnek is nehézséget okoz a két, egymás mellett a mondatban helyet foglaló főnévi csoport helyes felismerése – azaz hogy két külön vagy egy csoportot alkotnak-e (lásd az alábbi példákat).

A gyakorlatban használt n-gram modellek segítségével a szerkezet felismerése hagyományosan az AnaGramma elemzőrendszer működéséhez hasonlóan balról jobbra történik – egy fix méretű ablakban – az első (nem elhagyott) elem megtalálásával, és az első nem odaillő, nem a várt sorrendben álló vagy teljesen más típusú token (például központozás vagy finit ige) megtalálásával fejeződik be, egyfajta előretekintés eredményeképpen, melyről a későbbiekben még ejtek szót. Az n-gram modellek dolgát nehezíti, hogy a maximális főnévi csoportoknál a használt ablak szélessége sokszor kicsinek bizonyulhat. Ráadásul a főnévi csoportból bármelyik elem elhagyható úgy, hogy valamelyik elemnek mégis jelen kell lennie a mondatban azért, hogy jelölje a szerkezet helyét[1]. N-grammokkal nehezen modellezhető az is, amikor két egymás mellett lévő főnévi csoport határát kell meghatározni. Ilyenkor sok esetben mély szemantikai kategorizáció, nyelvérzék és nagyobb kontextus hiányában nem eldönthető, hogy két független vagy egy komplex, esetleg helytelenül írt elemről van-e szó. Hiába az elemek kötött sorrendje és a konstrukciós szabályok ismerete, az n-gram modellek nem tudnak jó jóslást adni a főbb szerkezetre sem, melyet a következő példákkal illusztrálok:

A (7) példában az első NP-nél a fej nem realizálódik, a második NP-nél hátravetett birtokos szerkezet van, ami a hagyományos n-gram ablakkal nem felismerhető. Így a címkéző program kimenete helytelenül egy NP.

[1]Itt nem tárgyalom a teljesen elhagyott szerkezetet, mert annak felismerése teljesen más módszert igényel.

DOI:10.15774/PPKE.ITK.2018.007

(7)	*A*	*legkisebb*	*húsvét*	*első*	*napján*	*nagyon*	*örült*	.
	DET	MN.SUPL.NOM	FN.NOM	MN.NOM	FN.PSE3.SUP	HAT	IGE.ME3	PUNCT
	[A	legkisebb]$_{\text{NP}}$	[húsvét	első	napján]$_{\text{NP}}$	nagyon	örült	.
	*[A	legkisebb	húsvét	első	napján]$_{\text{NP}}$	nagyon	örült	.
	*[A	legkisebb	húsvét]$_{\text{NP}}$	[első	napján]$_{\text{NP}}$	nagyon	örült	.

A (8) példában az első NP módosítóval bővített, a második NP egy birtokos szerkezet, melyben a birtokos nem hangzik el. A címkéző az ablak korlátai és a szemantikai tudás hiánya miatt könnyen összetéveszthető a negyedik sorban jelölt szerkezettel, ahol az első NP módosítóval bővített, továbbá a második módosítóval bővített NP birtokosával egy szerkezetet alkot. Az ilyen szerkezet nem fér bele a címkéző program ablakába.

(8)	*A*	*régi*	*ház*	*első*	*felesége*	*szerint*	*szép*	.
	DET	MN.NOM	FN.NOM	MN.NOM	FN.PSE3.NOM	NU	MN.NOM	PUNCT
	[A	régi	ház]$_{\text{NP}}$	[első	felesége	szerint]$_{\text{NP}}$	[szép]$_{\text{NP}}$	.
	*[A	régi	ház	első	felesége	szerint]$_{\text{NP}}$	[szép]$_{\text{NP}}$	.

A (9) példában az első NP feje – a főnév hiánya miatt – egy módosítóval, és az esete nominatívusz. A második, determinálatlan NP egy névutós szerkezet. A címkéző programnak döntenie kell, hogy az első NP-nek valóban a melléknév lesz-e a feje vagy inkább a következő főnév bővítménye lesz. Ebben az esetben még nem is vettük figyelembe a szófaji egyértelműsítés lehetséges hibáját az NP-felismeréskor.

(9)	*A*	*repülő*	*idő*	*előtt*	*leszállt*	.
	DET	FN.NOM	FN.NOM	NU	IGE.ME3	PUNCT
	[A	repülő]$_{\text{NP}}$	[idő	előtt]$_{\text{NP}}$	leszállt	.
	*DET	IGE.OKEP.NOM	FN.NOM	NU	IGE.ME3	PUNCT
	*[A	repülő	idő	előtt]$_{\text{NP}}$	leszállt	.

A (10a) példában látható, hogy az NP keresés feladata – a csővezeték miatt – a szófaji egyértelműsítésnek erősen ki van téve. Ugyanis a szemantikailag más értelmű, de (hibásan) azonos szófaji címkékkel ellátható a (10b) példa a gép szempontjából egyező a (10a) példával.

DOI:10.15774/PPKE.ITK.2018.007

(10) a.

Megerősített	*házakat*	*épített*	*Júdában*	.
IGE.MIB.NOM	FN.ACC	IGE.Me3	FN.INE	PUNCT
[Megerősített	házakat]$_{\text{NP}}$	épített	[Júdában]$_{\text{NP}}$	.

b.

Megerősített	*embereket*	*megrendült*	*hitükben*	.
*IGE.MIB.NOM	FN.ACC	IGE.Me3	FN.INE	PUNCT
*[Megerősített	embereket]$_{\text{NP}}$	megrendült	[hitükben]$_{\text{NP}}$	.
IGE.Me3	FN.ACC	IGE.MIB.NOM	FN.INE	PUNCT
Megerősített	[embereket]$_{\text{NP}}$	[megrendült	hitükben]$_{\text{NP}}$	.

Nem csak a csővezetékben lehet hiba, hiszen ha a szöveg nem felel meg a normáknak, akkor olyan szintaktikai többértelműség keletkezik. A *10 éves technika iránt érdeklődő fiúnak milyen ajándékot?* mondat első maxNP-je a (11) példában szemlélteti, hogy bár mindössze egy vessző hiányzik az értelmező szerkezetből, ez teljesen félreviheti az elemzést. Így a szöveg a második, rossznak jelölt értelmezést jelenti, de az emberi elemző számára a világismeret javítja a szerkezetet.

(11)

10	*éves*	*technika*	*iránt*	*érdeklődő*	*fiúnak*
SZN.NOM	MN.NOM	FN.NOM	NU	IGE.OKEP.NOM	FN.DAT
[10	éves]$_{\text{ADJP}}$	[[technika	iránt]$_{\text{ADJP}}$	érdeklődő]$_{\text{ADJP}}$	fiúnak
*[[10	éves	technika	iránt]$_{\text{NP}}$	érdeklődő]$_{\text{ADJP}}$	fiúnak

A (12) példában a konjunkció elemeinél nem tudhatja a címkéző program időben, hogy a konjunkció második fele is NP, ami így egy maximális NP-t alkot, mert a konjunkció jelölheti tagmondatok összekapcsolását is.

(12)

Jancsi	*és*	*Juliska*	*romlott*	*kishúga*	.
FN.NOM	KOT	FN.NOM	IGE.MIB.NOM	FN.PSE3.NOM	PUNCT
[Jancsi]$_{\text{NP}}$	és	[Juliska	romlott	kishúga]$_{\text{NP}}$	.

Az említett problémák egy részének orvoslása nem tartozik a dolgozatban megoldott feladatok közé, de elemzésükkel közelebb kerülhetünk a megoldáshoz. Az viszont egyértelműen látható, hogy a hosszabb szerkezetek helyes elemzéséhez

DOI:10.15774/PPKE.ITK.2018.007

egy, az ablak keretein túlmutató eszközre van szükség. Vagy az egész mondatot kell előre látni, hogy a szavak megkaphassák a megfelelő kontextusra vonatkozó információkat a pontos döntéshez, vagy pedig az egyes szavaknak kell ellenőrizniük – többnyire közvetlenül maguktól jobbra –, hogy nem tartoznak-e egy el nem hangzott vagy mondatban később következő szerkezethez. A következő fejezetekben az előbbi, míg az 5. fejezetben az utóbbi megközelítést vizsgálom.

2.2. Közvetlen részszerkezetek azonosítása mint szekvenciális címkézés

A hagyományos, az egész mondatot előre látó szekvenciális címkéző módszerekben közös, hogy a mondat egyes elemeihez a közvetlen környezetből, vagy akár a mondat összes többi eleméből származó, előre meghatározott szabályok szerint származtatott jellemzőket rendelnek. A program a címke és token együttes előfordulásának valószínűsége helyett az így nyert jellemzőkből – a kontextust is figyelembe véve – tudja az adott elemre vonatkozó címkeeloszlást (emissziós vagy unigram modell) kiszámolni például *maximum entrópia (ME)* módszer (Ratnaparkhi 1996) segítségével[1]. Ezen túl már csak a Markov tulajdonságot kell felhasználnia, hogy az 1.3.3. fejezetben már ismertetett képlet alapján működhessen. Az ilyen, az emissziós modelljében a maximum entrópia modellt és a címkeátmenet-modellben a Viterbi-algoritmust használó módszer az úgynevezett *maximum entrópia Markov modell (MEMM)* (McCallum, Freitag és Pereira 2000), melyet széleskörűen alkalmaznak változatos szekvenciális címkézési feladatokban.

Azért erőforrás-takarékosabb az egy vagy több tokent összefogó „zárójelezés" és adott esetben zárójelezett csoportokra különféle címkék aggatása, amikor több osztályt akarunk megkülönböztetni, mert alapvetően csak a tokenek zárójelekhez képesti pozícióját szeretnénk kódolni a tokenekhez hozzárendelt címkékkel. Ehhez a feladathoz – mivel nincsenek egymásba ágyazott zárójelek – viszont viszonylag kevés osztály megkülönböztetése is elég: meg kell különböztetnünk a

[1] A maximum entrópia modell kimenete egy valószínűségi becslés minden osztályra, amely arra a kérdésre válaszol, hogy „Milyen valószínűséggel tartozhat az adott elem az egyes osztályokba a másik osztályok helyett?". Látható, hogy a módszer nagy számú jellemzővel is elbír, de sok osztály esetén nagyon lelassul.

DOI:10.15774/PPKE.ITK.2018.007

nyitó és a berekesztő zárójelek mellett álló, az adott csoport első és utolsó tokenjét, valamint a csoport belsejében illetve az összes csoporton kívül álló „kilógó" (outlier) elemeket. A fenti különbségtétel mellett az utolsó osztálytól eltekintve hozzá kell adnunk az adott csoport elnevezését is a címkéhez, amennyiben több csoportról van szó.

Látható tehát, hogy ha az osztályozási feladatban minimálisra akarjuk csökkenteni az osztályok számát – a maximum entrópia algoritmus gyorsítása érdekében –, több lehetőségünk is van a reprezentációra, de mielőtt rátérnék a reprezentációk további részleteire, néhány példával szemléltetem, hogy mely tulajdonságban közös a szófaji egyértelműsítés, a közvetlen összetevős elemzés, a minimális és maximális főnévi csoport keresés valamint a névelem-felismerés feladata a reprezentáció szempontjából:

A (13) példában látható mondaton a szófaji egyértelműsítés címkézési feladatát szemléltetem zárójelezéssel, valamint az alternatív elemzést is feltüntetem. Mindig egy token kerül egy osztályba, és minden szót osztályozunk. Így a zárójelezés elhagyható (harmadik és ötödik sor). A program működése nagy vonalakban a környező lehetséges címkék sorozatainak kiértékelése n-gram modell segítségével, valamint a szavak és címkék a tanítóanyagbeli együttes előfordulásainak[1] vizsgálataival.

(13)

Légy	*a*	*feleségem*	*!*
[Légy]$_{\texttt{IGE.PE2}}$	[a]$_{\texttt{DET}}$	[feleségem]$_{\texttt{FN.PSE1.NOM}}$	[!]$_{\texttt{PUNCT}}$
IGE.PE2	DET	FN.PSE1.NOM	PUNCT
*[Légy]$_{\texttt{FN.NOM}}$	[a]$_{\texttt{DET}}$	[feleségem]$_{\texttt{FN.PSE1.NOM}}$	[!]$_{\texttt{PUNCT}}$
**FN.NOM*	DET	FN.PSE1.NOM	PUNCT

A minimális (14a) és maximális (14b) főnévi csoportok, valamint a közvetlen összetevős szerkezetek (14c) reprezentációja szembetűnően hasonló. Míg az első kettőben alapvetően egy osztály van, és minden más elem kívülálló, addig a

[1]Problémát jelentenek a tanítóanyagban nem szereplő OOV szavak, melyek kezelésére egy statisztikai ragozási modellt kell építeni, szabályalapú morfológiai elemzést adva vagy manuálisan kell meghatározni – ami a morfológia zárt lexikonja esetén nem teljes megoldás –, vagy pedig a felsorolt módszerek kombinációjával kell meghatározni a kívánt valószínűségi eloszlást, remélve, hogy a létrejövő címkesorozat-jelöltek közül a helyes lesz a legvalószínűbb.

DOI:10.15774/PPKE.ITK.2018.007

harmadik esetben sokkal több az osztály, és alig marad token, amely kívülálló. Tehát a különbségük egyedül az, hogy melyik elemeket jelöljük külön melyiektől, milyen felbontásban. A szófaji egyértelműsítéshez képest a módszerben az a különbség, hogy a szavak és címkék kapcsolatát bonyolult, mindkétoldali környezetből nyert jellemzők segítségével létrehozott valószínűségi modell alkotja, amely feladatspecifikus.

(14) a. *Legott* *a* *pilóta* *zöld* *autója* *után* *futott* .
HAT DET FN.NOM MN.NOM FN.PSE3.NOM NU IGE.ME3 PUNCT
[Legott]$_{\texttt{O}}$ [a pilóta]$_{\texttt{NP}}$ [zöld autója után]$_{\texttt{NP}}$ [futott]$_{\texttt{O}}$ [.]$_{\texttt{O}}$

MinNP keresés: a minimális NP-ket jelöljük. A többi elem O címkéjű.

b. *Legott* *a* *pilóta* *zöld* *autója* *után* *futott* .
HAT DET FN.NOM MN.NOM FN.PSE3.NOM NU IGE.ME3 PUNCT
[Legott]$_{\texttt{O}}$ [a pilóta zöld autója után]$_{\texttt{NP}}$ [futott]$_{\texttt{O}}$ [.]$_{\texttt{O}}$

MaxNP keresés: a maximális NP-ket jelöljük. A többi elem O címkéjű.

c. *Legott* *a* *pilóta* *zöld* *autója* *után* *futott* .
HAT DET FN.NOM MN.NOM FN.PSE3.NOM NU IGE.ME3 PUNCT
[Legott]$_{\texttt{ADVP}}$ [a pilóta zöld autója után]$_{\texttt{NP}}$ [futott]$_{\texttt{VP}}$ [.]$_{\texttt{O}}$

Közvetlen összetevők keresése: az összes közvetlen összetevőt jelöljük.

Az előbbiekkel rokon feladat a *névelem-felismerés* (15) is, mely esetében a névelemek típusa (dátum, szervezet, személy, helység, stb.) szerint zajlik az osztályozás, ezért sok kívülálló elem van, és ritkán fordul elő, hogy két névelem egymás mellé kerül. Ebben a feladatban is speciális, mindkétoldali környezetből és szótárakból nyert jellemzők által történik a szóhoz választandó legmegfelelőbb címke meghatározása.

DOI:10.15774/PPKE.ITK.2018.007

(15)

2006. június 15.-án	*Bill*	*Gates*	*bejelentette*	*,*	*hogy*
FN.SUP	FN.NOM	FN.NOM	IGE.ME3	PUNCT	KOT
[2006. június 15.-án]$_{\texttt{DATE}}$	[Bill	Gates]$_{\texttt{PERS}}$	bejelentette	,	hogy

lemond	*a*	*Microsoft*	*éléről*	*.*
IGE.e3	DET	FN.NOM	FN.DEL	PUNCT
lemond	a	[Microsoft]$_{\texttt{ORG}}$	éléről	.

Névelem-felismerés: a különböző típusú névelemek kapnak címkét. A többi elem `O` címkét kap.

Az összes fent ismertetett címkézési feladatban közös, hogy a kritikus pontjaik a szavak és a hozzájuk rendelhető címkék valószínűségi eloszlásának kiszámítása (emisszió), és hogy hány címkére kell felosztani az osztályozásunk terét (granularitás), mely döntésen a gyorsaság is nagyban múlik, hiszen a címkeátmenetek variabilitása elég kicsi ahhoz, hogy a tanítóanyagban kellő mennyiségű információ legyen elérhető[1]. A címkék száma sokszor csak a pontosság rovására csökkenthető – sőt gyakran finomabb felbontásra van szükség (lásd a 3. fejezet) –, mivel az csak a zárójelezés reprezentációjának megváltoztatásával oldható meg[2]. A következő fejezetben a különböző reprezentációk előnyeit és hátrányait ismertetem, – mivel megfigyeléseim szerint nagy hatással vannak a címkézőprogramok teljesítményére –, majd a későbbiekben rátérek a különböző címkéző programok összehasonlítására is.

2.3. A reprezentációk definíciói és különbségeik

Azokban az esetekben, ahol több token is kerülhet egy osztályba, a zárójelezés reprezentációja többféle módon is megvalósítható, attól függően, hogy milyen tulajdonságot tartunk fontosnak. Az egyes reprezentációknak vannak előnyei és hátrányai, melyeket ebben a fejezetben ismertetek.

[1] A zárójelezésre alapuló címkézéseknél ez triviálisan belátható, a szófaji egyértelműsítés feladatánál ez csak feltételezés.

[2] A szófaji egyértelműsítés esetében a teljesítmény lokálisan növelhető, ha például a felsőfokú melléknév és az igén lévő igekötő esetében nem foglalkozunk a szó elejével, mert így ezek rendre a fokozott melléknevek és az igekötő nélküli igék osztályával együtt kezelhetőek. Ez az egyszerűsítés az elemzés későbbi fázisában hátrányt okoz, amikor a nem jelölt igekötős igére rákötünk egy másik igéhez tartozó igekötőtőt, egy igekötőnek címkézett névmást vagy határozószót.

DOI:10.15774/PPKE.ITK.2018.007

A zárójelek reprezentációját `IOB` vagy `BIO` címkézésnek nevezik az elemeik angol neve alapján: *kezdő elem* (`B`, beginning, vagy `[`), *belső elem* (`I`, inside), *befejező vagy utolsó elem* (`E`, end vagy `L`, last vagy `]`), *külső elem* (`O`, outside vagy outlier), *egység hosszú elem* (`1` vagy `S`, single vagy `U`, uniqe vagy `[]`). A reprezentációk lehetnek kezdet-explicitek, vég-explicitek és teljesek aszerint, hogy az összes kezdő elem vagy befejező elem vagy mindkettő jelölve van-e. Ezen kívül lehetnek explicitek vagy implicitek aszerint, hogy a közvetlenül egymás után jövő, azonos osztályba tartozó elemek ugyanúgy vannak-e jelölve vagy nem.

Reprezentáció	Kezdet jelölve	Belső tokenek jelölve	Végződés jelölve	Explicit	Címkék száma
`IOB1`	Csak egymást követő csoportoknál	Igen	Nem	Nem	1+2*csoportok száma
`IOB2`, `BIO`	Igen	Igen	Nem	Igen	1+2*csoportok száma
`IOE1`	Nem	Igen	Csak egymást követő csoportoknál	Nem	1+2*csoportok száma
`IOE2`	Nem	Igen	Igen	Igen	1+2*csoportok száma
`IOBES`, `SBIEO`, `BILOU`, `IOBE1`	Igen	Igen	Igen	Igen	1+4*csoportok száma
`Kezdet-vég`	Igen	Nem	Igen	Igen	1+2*csoportok száma
`Kint-bent`	Nem	Igen	Nem	Igen	1+csoportok száma
`Zárójel nélküli`	Nem	Nem	Nem	Igen	1+csoportok száma

2.1. táblázat. A bemutatott `IOB` reprezentációk főbb tulajdonságai. Explicit egy jelölés, ha a sorrendjüktől függetlenül ugyanúgy jelöljük az azonos csoportokat.

A főbb reprezentációk tulajdonságait összefoglalja a 2.1. táblázat, a definícióik pedig a következők:

DOI:10.15774/PPKE.ITK.2018.007

- `IOB1, IOB2`: A külső (`O`) elemektől csoportonként megkülönböztetjük a belső elemet jelölő címkéket (`I`). Továbbá külön jelöljük a csoport kezdő elemét (`B`), `IOB2` jelölés esetén minden esetben, illetve `IOB1` jelölés esetén két egymást követő azonos típusú csoport esetén a másodiknál.

- `IOE1, IOE2`: A külső (`O`) elemektől csoportonként megkülönböztetjük a belső elemet jelölő címkéket (`I`). Továbbá külön jelöljük a csoport befejező elemét (`E`), `IOE2` jelölés esetén minden esetben, illetve `IOE1` jelölés esetén két egymást követő azonos típusú csoport esetén az elsőnél.

- `IOBES/SIBEO`: A külső (`O`) elemektől csoportonként megkülönböztetjük a belső elemet jelölő címkéket (`I`). Továbbá külön jelöljük a csoport kezdő (`B`) és befejező (`E`) elemét, valamint az egy tokenből álló elemet (`S`) is.

Vannak olyan reprezentációk is, amelyek speciálisak, és bár kevesebb címkével dolgoznak, nem képesek minden esetet kifejezni. Ilyenek a következők:

- `Kezdet-vég` jelölés: csak a kezdő (`B`), befejező (`E`) és az egység hosszú (`S`) címkéket tesszük ki, a belső elemeket jelölő (`I`) címkék hiányoznak. Hátránya, hogy egy rosszul címkézett elem helyrehozhatatlanul elrontja a zárójelezés helyességét.
- `Kint-bent` jelölés: csak azt jelöljük, hogy a csoporton belül (`I`) vagy kívül (`O`) helyezkedik el a token. Nem képes az egymás után jövő, azonos típusú elemek megkülönböztetésére.
- `Zárójel nélküli` jelölés: csak az osztályokat jelöljük. A fenti két jelölés mindkét hátrányos tulajdonságával rendelkezik, de speciális esetekben – például a szófaji egyértelműsítésnél, amikor nincs több tokenre kiterjedő elem – mégis ezt célszerű használni.

A különféle reprezentációkra ad egy példát a 2.2. táblázat. Természetesen a konverzió a különféle reprezentációk között nem triviális, és a 2.4.4. fejezetben bemutatott szavazási eljárás nagyban támaszkodik rá. A reprezentációk konverziójával kapcsolatos kutatásaimat a 3.4. fejezetben ismertetem.

DOI:10.15774/PPKE.ITK.2018.007

Token	IOB1	IOB2	IOE1	IOE2	IOBES	Kezdet -vég	Kint -bent	Zárójel nélküli
Tegnap	O	O	O	O	O	O	O	O
Bhutánba	I	B	I	E	S	S	I	NP
utazott	O	O	O	O	O	O	O	O
a	I	B	I	I	B	B	I	NP
magyar	I	I	I	I	I	O	I	NP
miniszterelnök	I	I	I	E	E	E	I	NP
,	O	O	O	O	O	O	O	O
ahol	O	O	O	O	O	O	O	O
a	I	B	I	I	B	B	I	NP
király	I	I	E	E	E	E	I	NP
nagyszabású	B	B	I	I	B	B	I	NP
ünnepséggel	I	I	I	E	E	E	I	NP
fogadta	O	O	O	O	O	O	O	O
.	O	O	O	O	O	O	O	O

2.2. táblázat. Példa az IOB reprezentációkra.

2.4. Gyakran használt címkézési eljárások

2.4.1. Jellemzőalapú eljárások magyar nyelvre

A magyar nyelvben a jelenleg használt algoritmusok a névelem-felismerés feladatában gyökereznek, és MEMM szekvenciális címkéző módszer (lásd a 2.2. fejezet) variánsai, melyekben közös, hogy képesek sok jellemzővel is boldogulni, viszont mivel az exponenciális osztályozó algoritmusok családjába tartoznak, ezért sok osztály esetén megengedhetetlenül lelassulnak. Hasonló történik túl sok jellemző bevezetése esetén, de még így is sokkal jobban testre szabható az algoritmus, mint a HMM.

Az algoritmust megvalósító *HunTag* program (Recski 2014; Recski és Varga 2009) a *HunNER* (Varga és Simon 2007) névelem-felismerő általánosított változata alapján készült, amely nem csak névelemeket, hanem tetszőleges szekvenciális címkézési feladatot – így közvetlen összetevős szerkezetek felismerését is – képes elvégezni[1]. Mivel a névelemek felismerésénél a teljesítmény szempontjából nem

[1] Az irodalomban nevezik az átmeneti verziót *HunChunk* néven is.

DOI:10.15774/PPKE.ITK.2018.007

számít, hogy bi- vagy trigrammokat használunk[1] (Simon 2013), ezért az általános HunTag programban csak a bigrammok kezelését implementálták. A főnévi csoportok felismerése is csak bigrammok használatával történt, és nem került sor a trigramalapú címkeátmenet-modell megvizsgálására.

Bár a két feladat azonos keretrendszert használ, és gyakorlatilag csak a jellemzőkészletekben különböznek (lásd 2.2. fejezet), nem szabad elfeledkeznünk róla, hogy a főnévi csoportok szintaktikailag motivált egységek, ezért jóval gyakoribbak, és gyakoriak közöttük az egymást követő (akár azonos típusú) elemek is, melyek határát is jól kell meghatározni (lásd 2.1. fejezet). Ezért ebből a szempontból jobban hasonlítanak a szófaji egyértelműsítéshez, ahol egyértelműen a trigrammok használata volt célravezető, továbbá az átmenetek jobban elnyúlhatnak, aminek kezelése nagyobb ablakot igényel. Szükséges volt megvizsgálnom tehát, hogy elérhető-e jobb eredmény a főnévi csoportok felismerésében a trigrammok használatával. Ehhez átírtam a HunTag programot és kiegészítettem az címkeátmenet-modelljét trigrammok kezelésével. A trigrammokkal kiegészített HunTag programot[2] *HunTag3*-nak neveztem el[3]. Mérési eredményeimet az angol nyelvű jellemzőalapú eljárások tárgyalása után ismertetem.

2.4.2. Jellemzőalapú eljárások angol nyelvre

A másik általam vizsgált rendszer, melyet azért választottam, mert angol nyelven kimagasló eredményeket ért el, a *Conditional Random Field (CRF)* (Lafferty, McCallum és Pereira 2001), a maximum entrópia módszerhez hasonlóan az exponenciális osztályozók családjába tartozó eljárás. Célja, hogy úgy kezelje a különböző megfigyeléseket, hogy végül egy konzisztens értelmezést alkosson belőlük. Az erőssége, hogy jól kezeli a különböző úgynevezett „grafikus jellemzők" csoportjait, amelyek a megfigyelt elem környezeteiből jönnek. Az elsőként a képfeldolgozásban alkalmazott módszer mára általános szekvenciális címkézési eljárássá vált. Hátránya, hogy sok osztály esetén a maximum entrópia módszerhez

[1] Azért nem számít, hogy bi- vagy trigrammokat használunk, mert a címkeátmenetek sokkal egyszerűbbek a ritkán egymás mellé kerülő, és ezért egy elemként értelmezhető csoportok miatt. Simon Eszter fenti eredményét méréssel is megerősítettem (Indig és Endrédy 2018).

[2] A *HunTag* programmal is végeztem méréseket az előző eredmények pontosabb reprodukálásához, de a módszer lényegében megegyezik a *HunTag3, bigram* változattal.

[3] `https://github.com/ppke-nlpg/huntag3`

DOI:10.15774/PPKE.ITK.2018.007

hasonlóan lelassul. A CRFsuite[1] (Okazaki 2007) egy gyors és jól használható implementációja az osztályozónak, bár a címkeátmenet-modellje csak bigrammokat képes kezelni.

A state-of-the-art eljárást csak a 2.4.4. fejezetben tárgyalom, mert az ott alkalmazott módszer nem jellemzőalapú, valamint az osztályok megsokszorozásával éri el a kívánt teljesítményt angolra, ezért magyar nyelvre nem alkalmazható – az alapvetően nagyságrendekkel magasabb számú szófaji címke osztály miatt –, és így közvetlenül nem összehasonlítható vele a két nyelv. A következő fejezetben a fent ismertetett jellemzőalapú módszereket hasonlítom össze mindkét nyelven.

2.4.3. Mérések angol és magyar nyelvre

A fent említett főbb módszerek (MEMM+bigram, MEMM+trigram, CRF) vizsgálatai angol és magyar nyelvre a csak nyelvenként a címkekészletek miatt eltérő, máskülönben azonos, nyelvfüggetlen jellemzőkészlettel történtek, hogy tisztán az algoritmusokat lehessen összehasonlítani. Paraméterállításra nem volt szükség, így nem volt a fejlesztéskor elkülönített adat sem. Magyar nyelvre a Szeged Treebank (Csendes, Csirik et al. 2005), angolra a CoNLL-2000 (Tjong Kim Sang és Buchholz 2000) korpuszon történt a mérés az irodalomban is használt felosztással.

A magyar nyelvű adatok az MSD (Erjavec 2010) és a KR (Kornai, Rebrus et al. 2004) morfológiai kódolással voltak annotálva, mivel a tanítóanyag ilyen formában volt elérhető. A KR kódrendszerben a morfoszintaktikai jegyek hierarchiája fagráfban elhelyezett bináris jegyekkel van kódolva, melynek linearizálásából jönnek létre címkék. Kifejezőerejét illetően úgy lett tervezve, hogy az MSD kódrendszerrel ekvivalens legyen, melyben a 10 hosszú kódokon belül az első pozíció adja meg a fő szófaji kategóriát, míg a további pozíciókban egyéb nyelvtani információk kódolódnak. Jellemző rájuk, hogy nehezen olvashatóak, mivel a KR kód nem tartalmazza az alapértelmezett jegyeket a rövidség miatt, az MSD kód pedig rövidítéseket tartalmaz, melyek feloldása az első elemtől függ. A felhasznált korpuszban több mint ezer különböző szófaji címke került felhasználásra, – mely nem fedi le az összes lehetőséget –, míg az angol nyelvű adat a *Penn Treebank* szófaji címkekészletével (Bies et al. 1995) van kódolva, mely összesen 36 címkéből

[1] `http://www.chokkan.org/software/crfsuite/`

DOI:10.15774/PPKE.ITK.2018.007

áll. A címkék számának majdnem két nagyságrendbeli különbsége mutatja a két feladat eltérő voltát.

Miháltz Márton végzett egy összehasonlító mérést a magyar nyelvre elérhető maximális főnévi csoport kereső módszerekkel, melyből az összehasonlításban csak a HunTag programmal végzett mérést tárgyalom (Miháltz 2011). A hivatkozott cikkekben szereplő mérések során a KR kód esetén azonos módszert használtak, Recski Gábor és Miháltz Márton mégis eltérő eredményre jutottak. Ezeknek az eredményét nem tudtam reprodukálni, így külön szerepeltetem őket[1] az általam Endrédy Istvánnal közösen végzett mérések (Endrédy és Indig 2015) számaitól, melyek tartalmazzák a hivatkozott módszerek (*„HunTag (alapvonal)"*) alapján előállt számokat is.

Feltüntetem továbbá a táblázatban a magyar *WordNetből* (Miháltz, Hatvani et al. 2008) vett jellemzőkkel kiegészített mérés eredményét (Endrédy és Indig 2015) is összehasonlításképpen, eszerint a magyar WordNetből vett szinonimákat jellemzőként használva javulás érhető el[2]. A magyar nyelvű eredmények a 2.3., az angol nyelvűek pedig a 2.4. táblázatban láthatóak.

	MSD	**KR**	**KR+WordNet**
T'n'T	68,52	70,95	-
Recski Gábor és Miháltz Márton eredményei (HunTag (alapvonal))	81,71	88,72/90,28	-
HunTag (alapvonal)	93,20	88,96	90,78
HunTag3, bigram	93,43	89,10	90,72
HunTag3, trigram	**93,59**	**89,83**	**91,50**
HunTag3, CRFsuite	92,27	89,12	89,77

2.3. táblázat. Magyar nyelvű eredmények a Szeged Treebanken, F-mértékben (%) megadva, különféle címkekészletekkel és módszerekkel. Fölül a hivatkozott cikkekből származó értékek, alul az Endrédy Istvánnal közös mérések (Endrédy és Indig 2015). A legjobb eredményt a trigram változat érte el.

[1] Ezen mérések módszere a reprodukált *„HunTag (alapvonal)"* mérésével azonos.

[2] A WordNetből vett jellemzők Endrédy István munkáját képezik.

DOI:10.15774/PPKE.ITK.2018.007

módszer	F-mérték (%)
HunTag	91,38
HunTag3, CRFsuite	**93,42**
HunTag3, bigram	92,79
HunTag3, trigram	93,41
SS05 (state-of-the-art)	94,01

2.4. táblázat. A HunTag3 program eredményei angol nyelvre a CoNLL-2000 adathalmazon. A state-of-the-art módszert is feltüntettem.

Látható, hogy az elvégzett mérések tükrében angolra a HunTag3-nál egy kicsivel jobban teljesített a CRF módszer, míg magyarra kettőjük közül a HunTag3 teljesített jobban. Ebből úgy tűnik, hogy nincs egy egységes módszer, mivel a nyelvek túlságosan különböznek egymástól, de – mivel a különbség nagyon kicsi – ennek megalapozott kijelentéséhez további mérések szükségesek. Jelen mérésből látszik, hogy a magyar nyelvű state-of-the-art módszer által elért eredményt, mely bigrammokat használt, meghaladta a trigram átmenetekkel működő módszer, viszont az angol state-of-the-art módszert egyik alkalmazott eljárással sem sikerült meghaladni[1], habár a trigrammok használata angol nyelven is teljesítménybeli javulást hozott. Az angol nyelvű korpuszon a CRF módszer segítségével trigram-átmenetmodellt használva nem végeztem méréseket a program hiányosságai miatt, viszont az angol nyelvű state-of-the-art módszer reprodukálásával és módosításával próbáltam javítani az angol címkézés minőségén, melyet terveim szerint a magyar nyelvű maximális NP keresés state-of-the-art módszerének javítására kívántam használni. A következő fejezetben az angol nyelvű state-of-the-art módszer reprodukálásának tanulságait ismertetem.

2.4.4. Az angol nyelvű state-of-the-art módszer reprodukálása

Kutatásaimban a továbbiakban az angol nyelvű közvetlen összetevők keresése feladatot state-of-the-art eredménnyel megoldó módszer, az *SS05 algoritmus* (Shen

[1]A módszert sajátkezűleg nem futtattam, a 2.4. táblázatban csak a cikkben ismertetett eredményt tárgyalom.

DOI:10.15774/PPKE.ITK.2018.007

és Sarkar 2005) megismerésére, reprodukálására és a magyar nyelvre történő adaptálására fókuszáltam. A módszer során a szerzők a T'n'T szófaji egyértelműsítésre fejlesztett HMM-alapú programot (Brants 2000) használták, kiegészítve egy lexikalizációs eljárással (Molina és Pla 2002) és a különböző `IOB` reprezentációkon betanított és tesztelt címkéző program kimenetein végzett egyszerű többségi szavazással.

A magyar nyelvre történő alkalmazásához a módszer legegyszerűbben átültethető komponensével a *különböző `IOB` reprezentációk közötti egyszerű többségi szavazással* keztem el foglalkozni. A meglévő címkéző programok segítségével egy kiterjesztett mérést végeztem, amivel azt akartam ellenőrizni, hogy a szavazás önmagában mennyire befolyásolja-e a címkézés teljesítményét.

A módszer reprodukálásához szükséges a *lexikalizációs eljárás* melynek lényege, hogy a címkekészletet finomabb felbontásúra alakítja át, melynek segítségével a címkéző program biztosabb döntést tud hozni. Az eredeti eljárásban (a későbbiekben *teljes* lexikalizáció néven hivatkozok rá) egy bizonyos küszöb fölötti gyakoriságú szavakhoz hozzárendeljük a szóalakot és a szófaji címkét is, míg a küszöb alatti szavak esetén csak a szófaji címkét. Mivel ez az eljárás nagyon sok különböző osztályt generál, a magyar nyelvre való adaptálhatóság végett módosítottam a lexikalizáción, hogy kevesebb osztály jöjjön létre[1]. Javasoltam egy *enyhe* lexikalizációs eljárást, ahol csak a küszöb fölötti szavak címkéjén változtatok, így csökkentve az osztályok számát. Az új, általam feltalált osztályozási eljárást és tulajdonságait a 3. fejezetben ismertetem részletesen.

Az *SS05 módszer* a fent bemutatott független komponensek együttes használatával működik, bár ezek a komponensek egymástól függetlenül nem lettek megvizsgálva. Ahhoz, hogy egyértelműen meghatározzam, hogy melyik komponens mennyit javít a teljesítményen más, egyszerűbb módszerekhez képest, valamint a részben vagy egészben magyar nyelvre történő adaptálhatóság igénye miatt, a rendelkezésemre álló összes osztályozóval, a lexikalizáció és a szavazás különféle kombinációival el kellett végeznem a mérést.

[1]Mérés közben így is arra jutottam, hogy a HunTag3 program a maximum entrópia eljárás miatt még így is meg nem engedhető módon lelassul a lexikalizáció miatti osztályszámnövekedéstől.

DOI:10.15774/PPKE.ITK.2018.007

Mivel a T'n'T program nem nyílt forráskódú – amely nehezítette a mérés reprodukálhatóságát –, próbáltam olyan címkéző programokat is megvizsgálni, amelyek azonos módszerrel működnek, viszont nyílt forrásúak és elérhetők a tudományos közösség számára[1]. Ezzel azt kívántam elérni, hogy ha sikerül reprodukálnom az eredményt, akkor a felhasznált programrendszert közzé tudjam tenni, mivel az eredetileg használt szoftver nem volt elérhető[2]. Így esett a választásom az NLTK programcsomag (Loper és Bird 2002) T'n'T implementációjára[3], valamint a már bemutatott PurePOS programra.

2.4.5. Eredmények

Külön lefuttattam a három lexikalizációs szint (*semmi, enyhe, teljes*) szerinti méréseket mindegyik programmal (T'n'T, NLTK-T'n'T, HunTag3-bigram, HunTag3-trigram, HunTag3-CRFsuite, CRFsuite a szerzője által ajánlott jellemzőkkel[4]) az öt főbb reprezentáción (`IOB1`, `IOB2`, `IOE1`, `IOE2`, `IOBES`) az ajánlott egyszerű többségi szavazással és anélkül. A szavazás önmagában vett hatását a 2.5. összefoglaló táblázat mutatja (a részletes eredményekért lásd az A. függelék).

Látható, hogy a T'n'T program különböző változatain kívül nincs igazi nyereség a szavazás következtében egyik reprezentáción vagy lexikalizációs szinten sem. Annak érdekében, hogy még jobb képet kapjak az egyszerű többségi szavazásról, megvizsgáltam a különböző címkézőprogramok eredményeinek szavazással történő javításának lehetőségét is a fent definiált lexikalizációs szintekkel együtt az összes reprezentáción, melyet a 2.6. táblázat mutat.

Az eredményeket összehasonlítva az A. függelékben található részletes eredményekkel látható, hogy a címkéző programok közötti szavazás is csak rontott a legjobb program eredményein. A 2.6. táblázatban kicsiben látható az az általános tendencia, hogy a lexikalizáció mint eljárás minden szinten nagymértékű javulást hozott, viszont kiemelkedően az *enyhe* lexikalizáció teljesített az eredetileg *teljes*

[1] A T'n'T program további korlátja, hogy 2048 címkénél többet nem képes kezelni, mely magyar nyelven nagyon hamar meghaladható a lexikalizáció alkalmazásával.

[2] Endrédy Istvánnak a szerzőkkel folytatott levelezése hatására megengedték, hogy publikáljuk a hozzánk eljuttatott, az általuk a cikkhez használt szoftvereket, melyekkel a mérés továbbra sem volt reprodukálható: `https://github.com/ppke-nlpg/SS05`

[3] `http://www.nltk.org/api/nltk.tag.html#nltk.tag.tnt.TnT`

[4] Ezt a módszert röviden „Hivatalos CRFsuite"-ként jelölöm a táblázatokban.

DOI:10.15774/PPKE.ITK.2018.007

	nincs	**enyhe**	**teljes**
T'n'T	1,908	0,284	0,542
NLTK T'n'T	1,868	0,262	0,462
PurePOS	-0,014	-0,056	0,39
HunTag3 Bigram	0,398		
HunTag3 Trigram	0,282		
HunTag3 CRFSuite	0,444	0,468	0,676
Hivatalos CRFSuite	0,23	0,27	0,466

2.5. táblázat. Az átlagos nyereség a szavazással összehasonlítva az összes reprezentáción az összes címkézőprogram esetén. A táblázat nem mutat igazi különbséget (<1%) a lexikalizált változatokon a T'n'T módszer változatait leszámítva.

lexikalizációval szemben. A reprezentációk esetén az `IOBES` reprezentáció volt a leghatékonyabb.

Ami a címkézőprogramokat illeti, egyértelműen két mezőny alakult ki: A HMM-alapú módszerek (T'n'T, NLTK-T'n'T, PurePOS) és a jellemző alapú módszerek (HunTag3 változatok és CRFsuite) csoportja. Ezen belül a *CRFsuite* program egyértelműen kiemelkedően teljesített, és csak ezzel sikerült az elvárt eredményeknek megfelelőt elérni. A T'n'T címkéző használatával a módszer reprodukálhatatlan volt. Mivel a *hivatalos CRFsuite* módszerrel még az eredetileg közölt, de nem reprodukálható értékeket is sikerült meghaladni, így arra kell következtetnem, hogy a T'n'T nem a legalkalmasabb címkéző erre a feladatra. Ennek tisztázására végzett méréseimet a következő fejezetekben fogom ismertetni.

Végeredményben kimondható, hogy az egyszerű többségi szavazás, amelyet a state-of-the-art módszer javasolt, nincs komoly hatással (<1%) az eredményekre, viszont sokat lassít a rendszeren, következésképpen fölösleges. Ugyanez igaz, ha a különböző vizsgált címkézők kimenetét szavaztatjuk egymás ellen (lásd a 2.6. táblázat). A lexikalizáció volt a fő javulási pont, különös tekintettel az általam javasolt lexikalizációs szintre, mely az eredeti lexikalizációs szintnél kevesebb osztályt tartalmaz, így gyorsítja a mérési folyamatot. A magyar nyelven is al-

DOI:10.15774/PPKE.ITK.2018.007

	lexikalizáció nélkül	**enyhe lexikalizáció**	**teljes lexikalizáció**
IOB1	91,89	*93,55*	93,45
IOB2	92,61	***94,11***	93,81
IOE1	92,00	*93,56*	93,51
IOE2	92,75	***94,36***	**94,04**
IOBES	93,32	***94,60***	**94,03**

2.6. táblázat. A különböző címkézők kimenetének szavaztatása minden reprezentáción, minden lexikalizációs szinten. A legjobb F-mértékek (%) pedig *dőlt betűvel* szedettek, valamint minden 94% fölötti F-mérték **félkövérrel** szedett. Minden reprezentáció esetén az *enyhe lexikalizáció* teljesített a legjobban.

kalmazott `IOBES` reprezentáció önmagában is sokkal jobb eredményt (kb. 1%) hozott a többi megvizsgált reprezentációval összehasonlítva, így a továbbiakban angolra is ezt érdemes használni. A vizsgált címkéző programok közül egyértelműen a T'n'T volt a leggyorsabb, viszont teljesítményben a CRFsuite egyértelműen meghaladta.

DOI:10.15774/PPKE.ITK.2018.007

2.5. Összefoglalás és kapcsolódó tézisek

A fejezetben bemutattam és összehasonlítottam a különböző címkézési feladatokat magyar nyelvre. Ismertettem azokat a nehézségeket, amelyek meghatározzák a jelenlegi kutatások irányát. Bemutattam, hogy hogyan oldhatók meg ezek a feladatok szekvenciális címkézéssel. Összehasonlítottam az ismert `IOB` reprezentációkat, ismertetve az előnyeiket és hátrányaikat. Angol nyelven megpróbáltam reprodukálni a közvetlen összetevők kereséséhez használt state-of-the-art módszer eredményeit, mely a különböző `IOB` reprezentációk egyszerű többségi szavazásán alapul, de kimutattam, hogy a javulás egyedül a lexikalizációnál tapasztalható.

1. Tézis. *Méréssel kimutattam, hogy nem helytálló az a szakirodalomból ismert állítás, amely szerint a különböző `IOB`-reprezentációk közötti szavazás szignifikáns javulást hoz az angol nyelvű főnévi csoportok meghatározásának minőségén.*

A tézist alátámasztó közlemények: [3]

Bár a fejezetben ismertetett angol nyelvű eredmények mélyebb betekintést engednek a szekvenciális címkézés ezen alkalmazásának működésébe – nem beszélve a további javítás lehetőségéről –, az ismertetett eredmények önállóan még nem voltak alkalmasak a magyar nyelvű főnévi csoportok keresésének további javítására. Viszont ettől függetlenül felismertem, hogy magyar nyelven a maximális NP-k keresésének feladatában a state-of-the-art módszer csak bigram címkeátmeneti modellt használ, mely eredménye javítható trigram címkeátmeneti modell használatával.

2. Tézis. *Az általam kifejlesztett HunTag3 program segítségével méréssel igazoltam (társszerzővel közösen), hogy a trigrammok használatával javulás érhető el a bigrammokhoz képest a magyar nyelvű maximális főnévi csoportok meghatározásában.*

A tézist alátámasztó közlemények: [8]

Az eredmények alkalmazhatósága elsősorban a magyar nyelvű maxNP keresés feladatára terjed ki, mivel egyéb szekvenciális címkézési feladatokra pl. a névelemek felismerésre nincs hatással.

DOI:10.15774/PPKE.ITK.2018.007

3. fejezet

Lexikalizációs eljárások

„Ne hagyd, hogy a megszokás csábítását biztonságnak érezd: ha egy hosszú időn át begyakorolt módszer használhatatlan lesz, akkor felkészületlenül állsz az új helyzet előtt."

(Isamu Fukui: Truancy)

3.1. A lexikalizációs eljárások célja és hatása

Lexikalizációs eljárásnak nevezem a továbbiakban mindazon eljárásokat, amelyek a szekvenciális címkézésben a minimálisan szükséges osztályokon túl újakat is bevezetnek az osztályozás idejére. Az eljárás célja az, hogy ezzel lehetővé váljon egy finomabb osztályozás a címkéző algoritmus számára. Az új osztályok létrehozásához pedig az adott elemek, a tokenek lexikális tulajdonságait használja fel[1].

Ezen finomabb osztályozások nagyon sokféleképpen jöhetnek létre, de közös bennük, hogy mindegyik célja a rendszer hatékonyságának növelése azáltal, hogy az osztályozó jobban el tudja különíteni az egyes finom, jól karakterizálható, kicsi osztályokat, mint a nagyobb fedésű, de nehezebben meghatározható osztályokat. Az eljárás bemenete az eredeti osztályozás, és a kimenet is ilyen formára lesz alakítva a kiértékeléshez, csak a belső működés történik a finomított osztályokon.

[1] A terminológiával nem értek egyet, de Molina és Pla (2002) definiálja ezen a néven az eredeti ötletet, így ezt a nevet használom én is.

DOI:10.15774/PPKE.ITK.2018.007

Bár a szerelőszalagban előrébb és hátrébb található elemek nem igénylik közvetlenül a finomabb osztályozás meglétét, az elvégzendő feladat, a főnévi csoportok és egyéb közvetlen összetevők felismerésének feladata során felmerült osztályok túl tágnak és általánosnak bizonyulhatnak[1] az osztályozó számára, ezzel rontva a teljesítményt. Ugyanakkor az is igaz, hogy a túl sok osztály az osztályozók sebességét nagyban rontja, ezért nem szabad túl sok osztályt sem létrehozni.

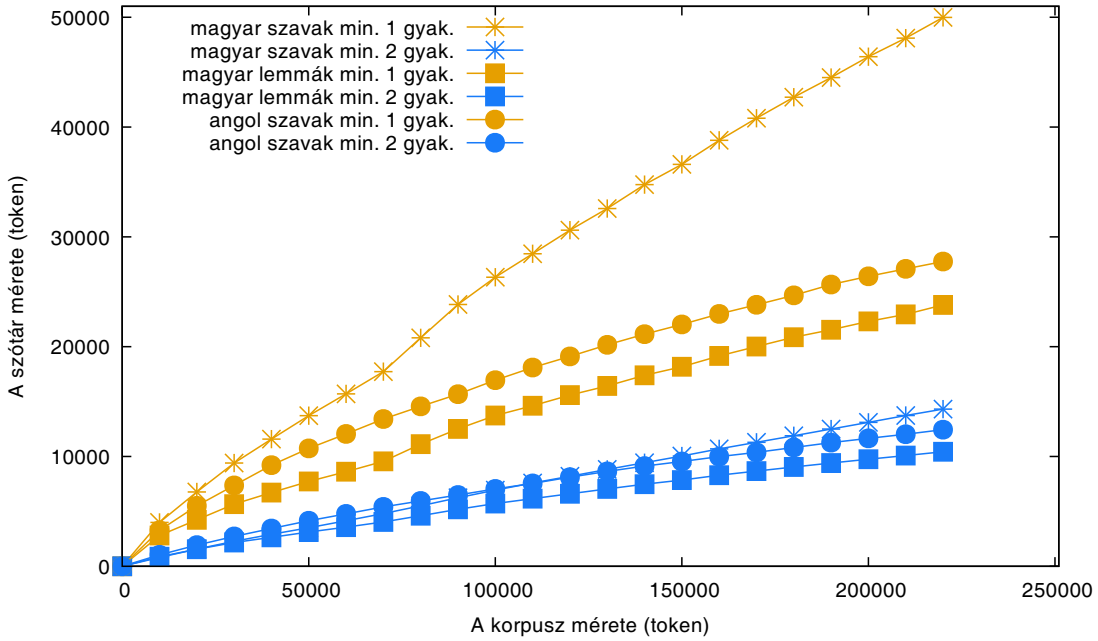
Egy példa a lexikalizációs eljárásra Recski Gábor eljárása (Recski és Varga 2012), amelyben a magyar főnévi csoportokat a hosszuk alapján osztályozta. A 2. fejezetben magyar nyelvre ezt az osztályozást alkalmaztam változtatás nélkül, de Gyalus Márk friss eredményei (Gyalus 2018) azt mutatják, hogy ezen megkülönböztetés nélkül tovább javítható a címkézés teljesítménye.

A dolgozatban bemutatott további vizsgálatok a szellemiségükben magyar nyelvű MaxNP keresés javítását célozzák, ámbátor jelen fejezetben a lexikalizációt csak angol tetszőleges közvetlen kifejezésekre, köztük minimális főnévi csoportokra alkalmaztam. A célom az volt, hogy a lexikalizáció tanulmányozásával a magyar nyelvű maximális főnévi csoportok felismerésének javításához merítsek ötleteket a jóval kiforrottabb angol nyelvű state-of-the-art megoldások vizsgálatával és továbbfejlesztésével. A bemutatott kutatásom célja tehát, hogy az angol nyelven jól működő módszerek jobban adaptálhatóak legyenek a magyar nyelv sajátosságaihoz.

A két nyelv közötti legfőbb különbség a lexikalizáció szempontjából az, hogy a magyar nyelvben nagyságrendekkel több szóalak van, melyeknek a szótövét is fontos figyelembe venni, ezzel szemben az angolban az egyszerű ragozási rendszer miatt erre nincs szükség, mivel az angol nyelv a ragozás helyett inkább a szavak sorrendjére támaszkodik. Ez az oka annak, hogy jó eredmények érhetőek el angol nyelven egyszerű trigram modellekkel, és kevésbé kell összpontosítani az egyes szavak tulajdonságaira. Az állításomat demonstrálandó, összehasonlítottam a CoNLL-2000 adathalmazt az MNSZ2 korpusz elejével abból a szempontból, hogy az adathalmaz növelésével hogyan változik a szótárméret (lásd a 3.1. ábra).

[1] A bizonytalanság abból fakad, hogy habár az adott modul szintjén javulás érhető el, amennyiben a megváltoztatott osztályozást megtartjuk a következő modulokban, akkor a 2.2. fejezet végén bemutatott esetben látható teljesítményromláshoz hasonló következhet be a szerelőszalag egészében.

DOI:10.15774/PPKE.ITK.2018.007



3.1. ábra. Az ábrán látható a korpusz méretének növelésével (x tengely, token) arányosan nő a szótár mérete (y tengely, token) angol és magyar nyelveken. Az angol szavak eloszlásának inkább a magyar lemmák felelnek meg. Ha csak a legalább kétszer előforduló szavakat számítjuk, akkor a két nyelv arányai igen hasonlóak.

A 3.1. ábrán látható, hogy a magyar nyelvre közvetlenül átültetve az angol nyelven egyszerűen a gyakori, többnyire funkciószavakból származtatott speciális osztályok képzése nem megvalósítható megoldás, mivel túl sok osztály jönne létre és megengedhetetlenül lelassítaná a címkéző működését. Ha viszont a szavak helyett azok lemmáját használnánk, melyek arányai jobban hasonlítanak az angol nyelvű szavakéhoz, akkor a lexikalizációs módszert át kell alakítani. Az átültetés megengedhetősége esetén sem biztos, hogy ugyan azt a hatást érnénk el, mint angol nyelven, mivel maguk a keresett csoportok is más felépítésűek. A következő fejezetben bemutatom, hogyan csökkentettem az osztályok számát, az általam vizsgált lexikalizációs eljárásokban a saját lexikalizációs módszeremmel angol nyelvre a sebesség és a teljesítmény javításával egyidejűleg, valamint a továbbiakban az összes paraméter hatására egyenként kierjedő vizsgálataim eredményét is ismertetnem.

DOI:10.15774/PPKE.ITK.2018.007

3.2. Az általam vizsgált lexikalizációs eljárások

A jelen dolgozatban tárgyalt eljárás eredeti változatát a HMM-ek teljesítményének javításához fejlesztették ki (Molina és Pla 2002). Az eljárás lényege, hogy a paraméterállításhoz használt halmazból kiválasztjuk a leggyakoribb szavakat, amelyek egy előre rögzített gyakorisági küszöb fölött vannak. Majd a tanulóhalmazban az ezekhez a szavakhoz használt címkékhez hozzáfűzzük magát a szót és a szófaji címkéjét is. A küszöb alatti szavak címkéi pedig csak a szavak szófaji címkéjét kapják meg (lásd a 3.1. táblázat). További megkötés, hogy a küszöb fölötti szavakat csak abban az esetben számítjuk gyakorinak, ha az osztályozni kívánt token csoportja benne van a leggyakoribb osztályokban[1].

	sima		**lexikalizált**			
	eredeti		***teljes***		***enyhe***	
token	**POS**	**IOB**	**POS**	**IOB**	**POS**	**IOB**
Tom	NNP	B-NP	NNP	NNP+B-NP	NNP	B-NP
said	VBD	O	VBD	O	VBD	O
the	DT	B-NP	the+DT	the+DT+B-NP	the+DT	the+DT+B-NP
truth	NN	I-NP	NN	NN+I-NP	NN	I-NP

3.1. táblázat. Különböző részletességű lexikalizációk: minden címkéhez hozzáfűzzük a szófaji címkét, és a gyakori tokenek címkéjéhez magát a tokent is (*teljes*), valamint az általam feltalált *enyhe* változat, ahol csak a küszöbnél gyakoribb tokeneken változtatunk. (Az irodalomban elterjedt jelöléssel ellentétben + szimbólumot használok elválasztóként, mert könnyebben kezelhető, mint az eredetileg használt - szimbólum.)

Az eljárás vitathatatlan érdeme, hogy az így keletkezett szó- és címkepárok, mivel gyakoriak, az osztályozó számára könnyen megtanulhatóak és elkülöníthetőek a többi osztálytól. A szövegben nagyjából egyenletesen eloszló speciális címkéket horgonyként tudja az osztályozó használni a címkeátmenet-modellben (erről bővebben lásd a 3.3.1. fejezetet). Így nem alakulhat ki hosszú, nehezen

[1]Ezek az eredeti cikkben a következő osztályok voltak: NP, PP, VP, ADVP

DOI:10.15774/PPKE.ITK.2018.007

felismerhető címkesorozat – mely jóval túllóg a trigram modell hatókörén –, valamint a gyakori és a ritka szavak gyakorlatilag „külön modellbe" kerülnek, ami javítja az osztályozás minőségét[1].

A 2.4.4. fejezetben ismertetett vizsgálatok után – melyben elvetettem az egyszerű többségi szavazást – az osztályok csökkentésével akartam az elért eredményt megtartani, és ennek tükrében változtattam a lexikalizációs eljáráson. Az elgondolásom szerint az osztályok számán legkönnyebben a ritka szavak esetében van értelme csökkenteni, mivel ezek – még ha összességében sokan is vannak – egyénileg számszerűen csak is kis számban vannak jelen, ennek következtében pedig célszerű minél szorosabb osztályozást alkalmazni, hogy a közös jellemzőik jobban érvényesülhessenek[2]. Az általam feltalált metódus, melyet *enyhe lexikalizációnak* neveztem el (lásd a 3.1. táblázat) csak a küszöb fölötti gyakori szavakra koncentrál, így a küszöb alatti ritka szavakat egyben kezeli. Az így létrejött „kevésbé finom" osztályozás megspórolja a ritka szavak szófaji címkék szerinti szétbontását, gyorsítja a címkéző működését, és a címkézés teljesítményét is javítja (a részletes eredményekhez lásd az A. függelék).

3.3. A lexikalizáció sarokpontjai

Az A. függelékben látható táblázatokból kiolvasható, hogy az *enyhe lexikalizáció* minden vizsgált konfigurációban túlszárnyalta a többi lexikalizációs szintet az összes reprezentációban, valamint a legjobb eredményt adó címkézőprogram a *hivatalos CRFsuite* volt. A további kutatást így ezzel a lexikalizációs eljárással és címkéző programmal folytattam az összes reprezentáción. A rendszerben így három meg nem vizsgált paraméter maradt: a küszöb, a lexikalizálandó szavak csoportjának típusaira történt megszorítás, valamint a lexikalizációhoz használt szavak forrása. A következő fejezetekben az ezekkel kapcsolatos eredményeimet ismertetem.

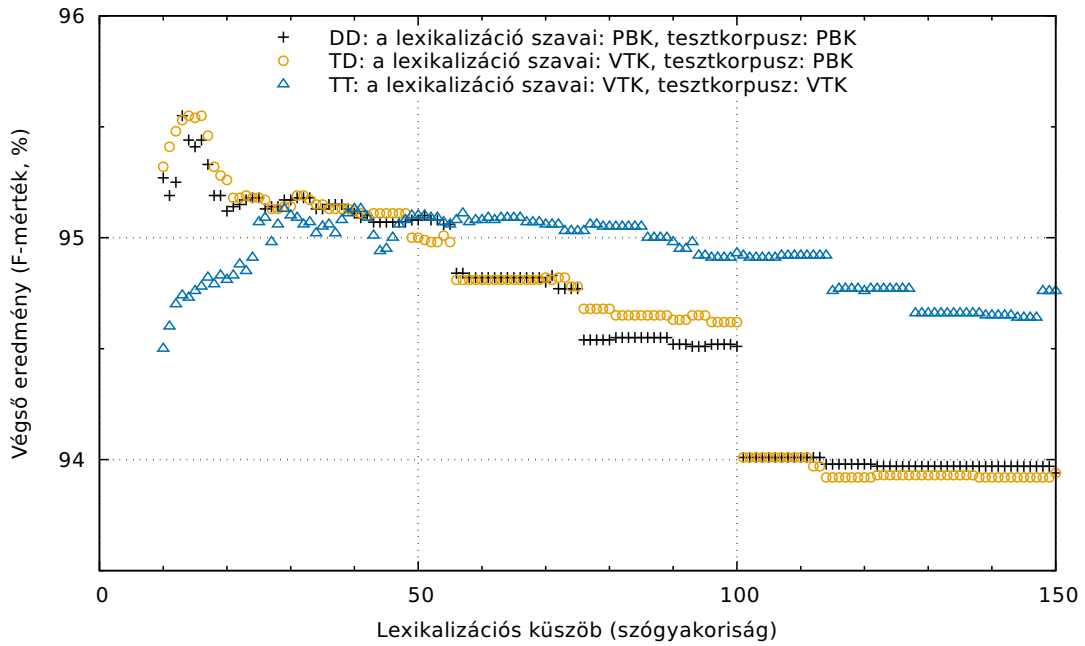
[1] Ez az eljárás nagyon hasonlít a T'n'T szófaji egyértelműsítőben használt, ismeretlen szavak kezelését biztosító modell működésére, ahol az algoritmus csak a bizonyos küszöbnél ritkább szavakból épít végződésfát a tanítóanyagban nem szereplő szavak kezelésére.

[2] A gyakori szavak esetén pedig a sok közös jellemző miatt érdemes az osztályozást finomítani, hogy elváljanak egymástól az elemek, és ezzel segítsék az átmeneti modellt.

DOI:10.15774/PPKE.ITK.2018.007

3.3.1. A küszöb

A lexikalizációhoz használt küszöbértéket az eredeti cikkben (Molina és Pla 2002) az ott rögzített körülményekhez mérten 50-ben húzták meg. Azok a tokenek kerültek a lexikalizálandó szavak halmazába, amelyek a paraméter állításához használt halmazban 50-nél gyakoribbak voltak. A 2.4.4. és a 3.2. fejezetekben tárgyalt vizsgálatok során (melyek eredménye az A. függelékben látható) ezt a paramétert alkalmaztam én is, hogy reprodukáljam az eredményeket, és megvizsgáljam az enyhe lexikalizáció teljesítményét. Azonban a megváltoztatott eljárás szükségessé tette, hogy felülvizsgáljam a küszöbszám értékének helyességét. Ezért a már meglévő kísérletező környezetemet úgy alakítottam át, hogy különböző küszöbértékekkel méréseket tudjak végezni, hogy a teljesítményt a küszöb értékének függvényében egy grafikonon ábrázolva össze tudjam hasonlítani. A minimális 10-ként választott küszöbtől indulva, mely a sok osztály miatt a még éppen megengedhető mértékben lassította le a számítást, egyesével növelve a küszöböt egészen addig haladtam, amíg el nem jutottam lexikalizálatlan formáig (lásd a 3.2. ábra).



3.2. ábra. Lexikalizációs küszöbök az `IOBES` reprezentáción (a többi reprezentáció hasonló, lásd a B. függelék) az ismertetett forgatókönyvek szerint. (PBK: paraméter beállításhoz használt korpusz, VTK: valódi teszt korpusz.)

DOI:10.15774/PPKE.ITK.2018.007

A 3.2. ábrán látható eredményeim nagyon látványosan igazolták a lexikalizáció pozitív hatását, mivel egészen alacsony küszöbértékig monoton egyenletes teljesítménynövekedés figyelhető meg. A `DD` forgatókönyv esetén – mely a hagyományos paraméteroptimalizálásis lépésnek felel meg – a 13-ra beállított küszöbértéknél a tanulóhalmazbeli tokenek 36,84%-a lexikalizálódik, így gyakorilatilag minden harmadik elem és a bigrammok 58%-a tartalmaz lexikalizált elemet. Tehát nem beszélhetünk arról, hogy csak a leggyakoribb funkciószavak lennének lexikalizálva. Viszont túl alacsony küszöbértéknél egy jól megfigyelhető visszaesés tapasztalható, mivel ebben az esetben bekerülnek olyan nagyon ritka szavak a rendszerbe, amelyekből a program nem tud kellően jó modellt építeni, és zavarólag hatnak. Felmerülhet továbbá az adathiány problémája, mivel ahogy a való életben is a paraméterállításhoz használt halmaz a tanítókorpusz szerves része (esetemben minden 10. mondata), a teszteléshez használt korpusz máshonnan származik és más tulajdonságai vannak. Mielőtt azonban a lexikalizáció forrását vizsgálnám, meg kell határozni, hogy a referenciaadat részét képező csoportok típusai szerinti szűrés mennyire fontos paraméter, hiszen ez az információ a teszthalmazból nem érhető el.

3.3.2. A lexikalizálandó szavak csoportjának típusai

Az előző fejezetekből látható a lexikalizáció forrásának hatása a címkézési folyamatra. Viszont mivel a teszthalmazból vett szavak esetében nem számíthatunk az őket tartalmazó csoportokra, meg kell vizsgálni a csoportok külön hatását is.

Ezt a mérést csak az `IOBES` reprezentáción végeztem, mivel az összes eddigi mérés során kiemelkedően teljesített. Csak azt vizsgáltam, hogy a csoportok szerinti szűrés be- és kikapcsolása miként hat a teljesítményre. Érdekes, a dolgozat témáján túlmutató kérdés, hogy a lehetséges csoportok kombinációiból melyik szerinti szűrés adhatná az optimális eredményt.

A szűrés hatása a tanítóanyagban úgy nyilvánul meg, hogy a lexikalizált szavak aránya a tanítóanyagban 36,84%-ról felmegy 52,46%-ra, valamint a címkézés sebessége jelentősen lelassul. A rendszer teljesítménye azonos küszöbérték mellett[1]

[1] A legjobban teljesítő küszöbérték kimérésére az módszer lasssúsága miatt nem történt meg.

DOI:10.15774/PPKE.ITK.2018.007

számottevően (kb. 2%), visszaesik 95,53%-ról 93,52%-ra. Ebből az következik, hogy egy nem elhanyagolható paraméterről van szó.

3.3.3. A lexikalizáció forrása

A fenti a tapasztalatokból kiindulva elkezdtem vizsgálni, hogy az immár sokkal alacsonyabban rögzített, legjobb teljesítményt produkáló küszöb esetén a teszthalmazból vett szavakkal nem érnék-e el jobb eredményt. Az eredeti módszerben a lexikalizálandó szavak a paraméterek állításához használt halmazból származtak, és így próbáltak általános modellként szolgálni tetszőleges teszthalmaz számára. Viszont ha a teszthalmaz előre ismert, és a feladat során a gyorsaságnál fontosabb a pontosság[1], akkor készíteni lehet egy speciális tanulóhalmazt, ami adaptálva van a teszthalmazhoz[2] úgy, hogy a teszthalmaz szavaiból – amik tesztelésidőben adottak[3] – vesszük a küszöb fölötti szavakat, hogy azok felhasználásával lexikalizálhassuk a tanítóhalmazt és így a modellt. Ilyen módon csökkenthető a tanítóanyag és a tesztanyag különbsége, és ezáltal javítható a módszer teljesítménye.

A végső kiértékelés előtt nincs rálátásunk a helyes megoldásra, ugyanakkor a módszer éles helyzetben ismeretlen adatra is alkalmazható[4], nem minősül a teszthalmazon tanításnak, valamint a túltanítás is elkerülhető, mivel mindig az adott teszthalmazra adaptált a modell, továbbá az így készített modellek teljesítménye összehasonlítható a különböző teszthalmazok között. A 3.2. ábrán látható a *valódi tesztkorpuszszal (VTK)* történt mérés, melyben az elméleti maximum meghatározására törekedtem[5]. Látható, hogy a maximum érték egy nagyobb küszöbértéken jelenik meg, viszont így sem éri el a *paraméter beállításához használt*

[1] Ilyenek lehetnek azon nagy korpuszok, amelyeket egyszer elemeznek a meglévő eszközökkel, és később mintákat és összefüggéseket keresnek bennük.

[2] A modell teszhalmazhoz történő adaptálása például a statisztikai gépi fordításban egy jól ismert és alkalmazott módszer (Farajian et al. 2017).

[3] Itt természetesen le kell mondani a csoportok típusa szerinti szűrésről, mivel azok a tesztelés közben nem ismertek.

[4] Ez esetben nem tudjuk kiválasztani a leggyakoribb csoportokat a lexikalizációhoz, ahogy az eredeti módszerben szerepelt, mivel nem ismertek a helyes címkék. Ezért csoporttól függetlenül mindent lexikalizálni kell.

[5] A mérésben felhasználtam a referenciaadatban szereplő csoportokat, hogy kiderítsem, milyen eredményt adna a rendszer az optimális lexikalizáció esetén.

DOI:10.15774/PPKE.ITK.2018.007

korpusz (PBK) szavaival történt mérések eredményét. Ha ezek után kikapcsolnánk a csoportok szűrését, hogy ne használjuk fel a referenciaadatot, további lassulással és teljesítményvesztéssel kell számolunk.

3.4. A struktúra ellenőrzése

Az `IOB` reprezentáció lényegében egy egy mélységű zárójelezési feladat, ahol az egyes zárójelekben a tartalom (álljon akár egy vagy több tokenből) egy címkét kap, amely megfelel a nemterminális állapotnak a további mélyebb elemzésben. Ezért fontos megjegyezni, hogy a `IOB` reprezentációval címkézett adat rendelkezik egy elengedhetetlen, mégis az irodalomban teljességgel mellőzött belső struktúrával, a helyes zárójelezéssel[1]. A kiértékelő programok ugyanis csak a helyesen zárójelezett tartalmat veszik figyelembe, a rosszul zárójelezett részeket eldobják és hibásnak minősítik.

Ennek a problémának különösen nagy a jelentősége akkor, ha az ember lexikalizációt vagy a `IOB` reprezentációk közötti konverziót szeretne végezni, hiszen a konverternek fel kell készülnie arra, hogy az adat nem jólformált. Egy olyan címkéző, amely nem képes megtartani a jólformáltságot, alkalmatlan arra, hogy a kimenetén lexikalizációt vagy `IOB` reprezentáció konverziót hajtsanak végre[2].

Vegyük észre, hogy amennyiben bármelyik fenti módszert is alkalmazzuk – vagy akár mindkettőt egyszerre, mint az előző state-of-the-art módszer esetén (lásd a 2.4.4. fejezet) –, emiatt több osztály közül kell választania a címkéző programnak. Viszont az osztályok számának növelésével együtt nő a rosszul formált elemek száma, mivel a címkéző átmeneti modellje a globális struktúra ismerete nélkül nem képes megtanulni adat hiányában a ritka átmeneteket. Ezért különösen fontos a jó címkéző algoritmus kiválasztása. Amennyiben a 2.4.4. fejezetben bemutatott eljárásban használt, egy különböző reprezentációkon tanított címkéző kimenetein történő szavazást alkalmazó módszert szeretnénk használni, akkor egy olyan `IOB` reprezentációk közötti konverziót elvégezni tudó programra van szükségünk, amely akár nem jól formált zárójelezés esetén is helyes eredményt ad, például a kimenet jólformálttá alakításával.

[1] Kivéve, ahol minden token külön csoportot alkot (pl. szófaji egyértelműsítés).

[2] Ilyen címkéző a T'n'T, amelyet ebből a szempontból a 2.4.5. fejezetben vizsgáltam.

DOI:10.15774/PPKE.ITK.2018.007

3.4.1. Metrika a szekvenciális címkézők osztályozására

A jólformáltság ellenőrzésére létrehoztam egy mérőszámot, amely azt mutatja meg, hogy hány címkét kell minimálisan megváltoztatunk ahhoz, hogy a teljes adaton jólformált zárójelezést kapjunk. Ez a mérőszám alkalmas a különböző címkézési módszerek minősítésére és rangsorolására. Ne feledjük, hogy a kapott érték csak egy közelítő adat, mivel nem tudjuk az összes elképzelhető szövegen lefuttatni a címkéző módszereket, viszont azonos (sztenderd) adaton futtatva viszonyszámként alkalmazható a különböző címkéző programok és eljárások képességeinek minősítésére, tehát hogy mennyire képesek megtartani az adaton a zárójelezés jólformáltságát.

Érdemes még megfigyelni, hogy bár egyes reprezentációknál a gyakorlatban jóval nagyobb ez a szám, mint másoknál, ez nem jelenti azt, hogy azok címkézése rosszabb lenne a többihez képest. Azért van ez, mert bizonyos nem explicit reprezentációknál egy jólformált sorozat egy másik jólformált sorozattá módosul, és így a továbbiakban nem javítható. Ha feltételezzük, hogy a megfelelő javító algoritmus birtokában az összes elromlott címkesorozatot a helyes címkére tudjuk javítani, akkor az a kiértékelésnél komoly előnyt jelenthet, nem beszélve a későbbi programok – amelyek nem biztos, hogy fel vannak készítve a rosszulformált bemenetre – működésének segítéséről. Természetesen a jólformáltság nem jelent közvetlenül helyességet, de méréskor nem érdemes így veszni hagyni a javítható címkéket, főleg akkor, ha nagy számban fordulnak elő.

3.4.2. Az `IOB` konverterek alkalmassága a jólformáltság javítására

Ahhoz, hogy az elméletet gyakorlatba ültessem, megvizsgáltam az elérhető `IOB`-reprezentáció konvertereket. A 2.4.4. fejezetben bemutatott előző state-of-the-art módszer, az *SS05* szerzői által használt szoftverek reprezentáció konvertáló modulja több súlyos hibától szenvedett[1], így használhatatlan volt a célra, viszont megmutatta a probléma valódi fontosságát. A program elve a reprezentáció mátrix minden elemének implementálása volt. Sok duplikációt találtam a kódban, ezek számos hibára adtak lehetőséget.

[1] `https://github.com/ppke-nlpg/SS05/issues`

DOI:10.15774/PPKE.ITK.2018.007

Az egyetlen szabadon elérhető IOB-reprezentáció konverter a Christopher Manning által írt Stanford CoreNLP szoftvercsomag (Christopher D. Manning et al. 2014) részét képező *IOBUtils*-ban található konverter[1] volt. A programban használt módszer a zárójelezés reprezentációjától függetlenül az egymás mellett lévő, azonos csoportba tartozó tokenek felismerésén, majd megfelelő reprezentációban való kiírásán alapul, megspórolva a konverziós mátrix felét. A program alapvető működéséből fakadóan javítja a jólformáltsági hibákat a konverzió során, ezért rosszul formált adaton is működik, mindig jólformált kimenetet adva. A program paraméterezhető úgy, hogy a kimeneti és a bemeneti reprezentáció is megegyezzen, és ilyenkor a megváltoztatott címkék száma jelenti a jólformáltság eléréséhez szükséges viszonyszámot, amelyet össze lehet hasonlítani más címkézőprogramok kimenetével, más reprezentációkkal vagy más lexikalizációs módszerekkel.

3.4.3. A címkéző és a lexikalizáció hatása a jólformáltságra

Megvizsgáltam, hogy a 2. fejezetben ismertetett címkézőprogramok mennyire tartják meg a jólformált bemeneti adaton a helyes zárójelezettséget a vizsgált lexikalizációs eljárások mellett a különböző IOB-reprezentációkon. A 3.2 táblázatban látható, hogy teljes lexikalizáció esetén az osztályok száma miatt megnőtt a rosszul formált zárójelezések száma a legjobban teljesítő címkézők esetén is.

	T'n'T	**NLTK T'n'T**	***HunTag3* CRFSuite**	**Hivatalos CRFSuite**	**PurePOS**
IOB1	168/234/319	148/230/313	286/260/266	306/294/304	197/274/317
IOB2	423/662/634	490/658/633	0/19/168	0/14/111	0/4/49
IOE1	0/1/1	0/2/2	4/13/11	0/2/2	0/0/0
IOE2	174/107/205	187/84/215	0/44/254	0/12/158	3/16/76
IOBES	862/805/985	647/702/898	2/95/865	2/51/521	2/22/210

3.2. táblázat. A rosszul formált címkesorozatok száma (nincs/enyhe/teljes lexikalizáció esetén): megfigyelhetjük, hogy programonként, reprezentációnként és lexikalizációs szintenként nagy mértékben eltérnek a számok, de az egyértelműen látszik, hogy a lexikalizáció mélyítése nehezíti a programok számára a jólformált címkeátmenetek megtartását.

[1] `https://github.com/stanfordnlp/CoreNLP/blob/master/src/edu/stanford/nlp/sequences/IOBUtils.java`

DOI:10.15774/PPKE.ITK.2018.007

Továbbá megfigyelhető a táblázatban, hogy a 2.4.4. fejezetben bemutatott előző state-of-the-art módszer lényege, a különböző `IOB` reprezentációkon tanított címkéző kimenetének szavazással történő egyértelműsítése csak a rosszul formált címkesorozatok magas száma esetén tud valós teljesítménynövekedést hozni. A módszer magával vonja azt, hogy minden felhasznált reprezentációról minden további reprezentációba konvertálás történjen, ami a rosszul formált adat és az erre nem felkészített konverter ersetén katasztrófális eredményt ad. Az eredeti cikkben egy olyan címkéző program közbeiktatásával (T'n'T) történt a mérés, amely képtelen volt megtartani a jólformáltságot, így a szerzők által használt, a rosszul formált adatra nem felkészített konvertereknek így nem volt lehetősége valódi eredményt produkálni. Erre a szerzőkkel történő együttműködés során jöttünk rá Endrédy István kollégámmal, amikor megkaptuk a szerzőktől az eredeti kódot[1]. Az eredmény nem csak reprodukálhatatlan volt, hanem teljességgel műtermék. Az ilyen hibák jövőbeni elkerülése végett állítottam fel az ismertetett metrikát a címkéző programok jólformáltság megtartási képességének vizsgálatára és eszerint történő osztályozásukra.

3.5. Következtetések

A fenti állítások helyességét ellenőrizendő, több felosztásban is megvizsgáltam a rendszer működését. A CoNLL-2000 adathalmaz tanítóanyagából a hagyományosan minden 10. mondat helyett tízféleképpen vettem a mondatokat, és az így keletkezett tíz felosztást különböző méretű teszthalmazokon mértem ki. Ehhez a teszthalmazt először feleztem, majd negyedeltem, és az így keletkezett, különböző méretű és tartalmú halmazokon (összesen hét darabon) megismételtem a méréseket, melyek eredményét összefoglalva a C. függelék mutatja.

Bár az enyhe lexikalizációra, alacsonyabb, 13-as küszöbre és a CRF címkézőre épülő angol nyelvű eredményeim a közvetlen összetevők és minNP-k keresésének feladatában jóval meghaladták a state-of-the-art módszer (HMM, teljes lexikalizáció, 50-es küszöb, reprezentációk közötti szavazás) teljesítményét (lásd a 3.3.

[1] `https://github.com/ppke-nlpg/SS05`

DOI:10.15774/PPKE.ITK.2018.007

táblázat), sajnos angol nyelven ez a módszer kevésbé alkalmazható, mivel napjainkra már elég jó minőségű szintaktikai elemzők állnak rendelkezésre angol nyelvre, amelyek jobban képesek ellátni az eredeti feladatot, azaz a főbb szintaktikai csoportok és viszonyaik azonosítását, mint a szekvenciális címkézés.

módszer	**közvetlen összetevők**	**főnévi csoportok**
Shen és Sarkar (2005)	94,01	95,23
Indig és Endrédy (2018), 50-es küszöbbel	95,06	96,49
Indig (2017), a 13-as küszöbbel	**95,53**	**96,69**

3.3. táblázat. A végső eredmények összefoglalása (F-mérték, %), amelyek meghaladják a state-of-the-art módszer eredményét.

Az eredményekből látható, hogy az angol nyelvben a közvetlen összetevők és a minimális főnévi csoportok megtalálása tulajdonképpen közelebb áll a szófaji egyértelműsítéshez, mint eddig gondoltuk. Majdnem minden tokenhez külön osztályt kell rendelnünk, amely a szófaji egyértelműsítésen túl a közvetlen környezetéből képes egyértelműsíteni a lehetséges `IOB` címkéjét is. A probléma csak az igazán ritka szavakkal van, melyek viszont sokfélék. Angol nyelv esetén ez nehéz, hiszen kevés szófaji címkével leírhatóak a szavak. Tehát érdemes lenne párhuzamosan végezni a szófaji egyértelműsítést és a közvetlen összetevők megtalálását, abban a reményben, hogy ebből mindkét módszer profitál[1].

Továbbá az is látható, hogy magyar nyelvre nem alkalmazható közvetlenül a továbbfejlesztett módszer sem, mivel alapvetően a szófaji címkék számossága is egy nagyságrenddel nagyobb, ami már csak a ritka szavakat tekintve is túlzottan lelassítja a rendszert. Viszont az ötlet, hogy a meglévő szófaji egyértelműsítőben (amely HMM-alapú, és így elbír sok osztállyal is) a szófaji egyértelműsítéssel párhuzamosan a közvetlen összetevők, így a főnévi csoportok felismerése is megtörténjen, egy ígéretes kutatási irány. A létrehozott lexikalizációs módszer célja tehát az, hogy olyan, morfológiában nem túl gazdag nyelveken, ahol még nincs elég jó szintaktikai elemző rendszer, ötletet adjon a közvetlen összetevők felismerését célzó rendszer teljesítményének növelésére.

[1] A programozási nyelveknél a szintaktikai elemzőbe szokták becsomagolni a lexikális elemzőt, ami így csak egy újabb szintté válik a szintaxis fában, egyszerűsítve a karbantartást. Ezt az eljárást *scannerless parsing* néven ismeri az irodalom (Visser 1997).

DOI:10.15774/PPKE.ITK.2018.007

3.6. Összefoglalás és kapcsolódó tézisek

Bemutattam az általam vizsgált lexikalizációs eljárások működését és hatását. Az általam feltalált lexikalizációs eljárást az angol nyelvű közvetlen összetevők keresésének feladatán vizsgáltam meg, ugyanakkor más nyelveken és feladatokon is alkalmazható.

3. Tézis. *Létrehoztam egy új, általános, szekvenciális címkézésre alkalmazható lexikalizációs eljárást, melynek első konkrét alkalmazása tetszőleges részszerkezetek hatékony azonosítását szolgálja.*

A tézist alátámasztó közlemények: [2, 3]

Az általam feltalált lexikalizációs eljárással és az optimális küszöbérték meghatározásával és alkalmazásával meghaladtam az angol nyelvű közvetlen összetevős keresés feladatán a state-of-the-art módszer teljesítményét.

4. Tézis. *Az általam kidolgozott eljárás angol nyelvű főnévi csoportokra méréssel igazolhatóan felülmúlja a jelenleg ismert módszerek F-mértékét.*

A tézist alátámasztó közlemények: [2, 3]

Bemutattam, hogy mennyire fontos az `IOB`-reprezentációk konverziójánál a megfelelően felkészített konverter alkalmazása, valamint az, hogy a címkéző program fenn tudja tartani a jólformáltságot a kimeneti címkesorozatok zárójelezésében. Ennek mérésére kidolgoztam egy metrikát, amit gyakorlatban alkalmaztam az angol nyelvű közvetlen összetevők keresésének feladatán.

5. Tézis. *Kidolgoztam egy zárójelezési módszert, mely egyfajta metrikaként a címkézési feladatra készített módszereket minőség szerint rendezni tudja.*

A tézist alátámasztó közlemények: [2, 3]

DOI:10.15774/PPKE.ITK.2018.007

4. fejezet

Erőforrások összekapcsolása

„Ha meg tudsz nézni valamit a saját szemeddel, akkor nincs szükséged arra, hogy mások véleményére hallgass."
(Takami Kósun: Battle Royale)

4.1. Az erőforrások összekapcsolásának célja

Tim Berners Lee, a *szemantikus web* feltalálója azt gondolta, hogy majd lesz olyan része az internetnek, ahol a gépek szemantikus lekérdezéseket tudnak lebonyolítani egymással emberi interakció nélkül (Berners-Lee, Hendler és Lassila 2001). Az elképzelése magában hordozta, hogy az érintett weboldalak olyan módon vannak megírva, hogy szabványosan tudjanak kommunikálni egymással. Ahhoz, hogy ez megtörténjen, nagyon nagy humánerőforrás-befektetés lett volna szükséges a weboldalak készítőinek részéről, ezért a projekt új irányt vett. Az interneten az addigra nagyon megszaporodott szabadon elérhető adatbázisokat kezdték összekapcsolni úgy, hogy komplex, szemantikus lekérdezéseket tudjanak rajtuk végrehajtani. Ebből a gondolatból lett a *Linked Open Data mozgalom*, melyet a *W3C* is támogat[1]. Az ötletet továbbgondolták, és elkezdték összekötni a nyelvtechnológiában használt szemantikus információt tartalmazó erőforrásokat. Ilyenek voltak a különféle *WordNetek* és a *SemLink* projekt„ melyeket a későbbiekben részletezek.

[1] `http://linkeddata.org/`

DOI:10.15774/PPKE.ITK.2018.007

A dolgozat szempontjából az erőforrások összekapcsolása azért érdekes, mert magyar nyelvre rendelkezésre állnak nagy fedésű igei vonzatkeret-adatbázisok – melyek egyike egy szabályalapú gépi fordítórendszer része lévén rendelkezik keretenkénti angol nyelvű megfeleltetéssel (lásd a 4.3.1. fejezet) –, ezek viszont nem rendelkeznek szemantikai annotációval, mely már valójában nyelvfüggetlen[1] és az elemzéshez felhasználható. Angol nyelven viszont számtalan jó minőségű erőforrás rendelkezésre áll, melyek tartalmaznak nyelvfüggetlen, szemantikus reprezentációra vonatkozó adatokat. Ezen annotációk magyar nyelvre történő előállítása költséges és emberierőforrás-pazarlás lenne, valamint teljességgel duplikálná a meglévő lexikális erőforrásokat. Kézenfekvő volt tehát a magyar nyelvre elérhető, angol nyelvvel keretenként összekötött, magyar-angol kétnyelvű vonzatkeret leírások összekapcsolása az angol nyelven elérhető széleskörű erőforrásokkal, hogy az így létrehozott kapcsolatok alapján bővítsük nyelvfüggetlen szemantikai információval a magyar nyelvű erőforrásokat. A következő fejezetekben néhány példát mutatok az összekapcsolt erőforrásokra, majd rátérek a dolgozat szempontjából érdekes erőforrások bemutatására.

4.2. Meglévő összekapcsolt erőforrások

4.2.1. Lexikális ontológiák

Az igei vonzatkeretek összekapcsolását segíthetik a lexikális ontológiák, melyek az ontológiák azon alcsoportjába tartoznak, amelyek a nyelvből, és azon belül is a szavakból és különféle jelentéseikből indulnak ki. A gyakorlatban az ilyen ontológiák úgy néznek ki, hogy a szavak egyes jelentései és velük szinonim elemek egy halmazba vannak sorolva, melyet *synsetnek* neveznek, és a synseteket összekötik különféle hasonlósági viszonyok – melyek az emberi agy működését mintázva készültek –, melyek kifejezhetik az alá-fölérendeltséget és az ellentét viszonyt is.

Az alá-fölérendeltségi viszony a szemantikai tartalmazást jelenti: ha egy fogalom bővebb, absztraktabb és magában foglal több konkrétabb fogalmat, akkor az

[1]Bár a két nyelv lexikális felbontása nem feltétlenül egyezik – itt az E/3-ben szétválasztott nemektől egészen a „lóöszvér” és „szamáröszvér” megkülönböztetéséig terjedhet a skála (mivel az angol *mule* a hím szamár és kanca ló kereszteződését, míg a *hinny* a nőstény szamár és mén ló kereszteződését jelenti) –, a főbb logikai viszonyok – mint amilyenek a tematikus szerepek – a lexikon túlnyomó részének esetében megegyeznek.

DOI:10.15774/PPKE.ITK.2018.007

előbbi az utóbbiak fölé lesz rendelve. Az így kialakult hálón barangolva lehet következtetéseket levonni, hogy két távoli rokonságban lévő szó milyen viszonyban van egymással[1]. A következő fejezetekben látni fogjuk, hogy az ilyen ontológiák néha nyelvek közötti kapcsolatokat is tartalmaznak, melyek az összekapcsolt erőforrások előfutárainak számítanak. Ezek a kapcsolatok kiválóan használhatók az igei vonzatkeret-adatbázisok azonos jelentésű elemeinek megtalálásában és a homonimák egyértelműsítésében.

4.2.1.1. Princeton WordNet

A Princeton WordNet (Miller 1995) az első olyan lexikális ontológia, amely az angol nyelv felhasználásával a szavak jelentése alapján próbálta „felépíteni a világot". Jelenleg a 3.1-es változata érhető el online,[2] mely 155 287 darab szót tartalmaz 117 659 darab synsetben, összesen 206 941 darab önálló jelentéssel. A szavak jelentései mellé definíció is meg van adva, így szótárként illetve tezauruszként is használható. Csak négy szófaji kategóriát tartalmaz (ideértve a többszavas kifejezéseket): főnév, melléknév, ige és határozószó. Azon szavakat, amelyek nem tartoznak ezekbe a kategóriákba, egyáltalán nem tartalmazza.

A kezdeményezés alapján sokan sok irányban próbáltak hasonló adatbázisokat létrehozni, melyek más nyelveket is támogattak. Ilyen volt például az EuroWordNet. Mára a vektoros modellek elterjedésével a szerepük marginalizálódott, mivel a nyelvek közötti kapcsolatokat azok a gép számára hatékonyabban tudják reprezentálni (Handler 2014).

4.2.1.2. EuroWordNet

Az *EuroWordNet* (Vossen et al. 1998) a nyugat-európai országok kezdeményezéseként jött létre. A célja az volt, hogy a főbb európai nyelveken (holland, olasz, spanyol, német, francia, cseh és észt) egy olyan WordNetet hozzanak létre, amely a nyelvek egyéni tulajdonságaihoz igazítja az egyes al-WordNeteket, de eközben tartalmaz nyelvek közötti kapcsolatokat is, amelyek segítségével a háló még részletesebb és több funkciójú lesz.

[1] A hálón történő lépegetéshez kifejlesztettek többféle metrikát, amelyek segítségével két jelentés hasonlósága számszerűen meghatározható (Pedersen, Patwardhan és Michelizzi 2004).

[2] `http://wordnetweb.princeton.edu/perl/webwn`

DOI:10.15774/PPKE.ITK.2018.007

Ahhoz, hogy az egyes nyelvek szabványosan legyenek leírva, létrehozták az úgynevezett felső ontológiát, mely az alapvető fogalmakat ábrázolja nyelvfüggetlen módon. Később a VerbIndex szemantikus leképezéseinek ez az ontológia adta az alapját (Schuler 2005).

4.2.1.3. Magyar WordNet

A *Magyar WordNet* (Miháltz, Hatvani et al. 2008; Prószéky, Miháltz és Kuti 2013) az MTA Nyelvtudományi Intézet, a Szegedi Egyetem Informatikai tanszékcsoportja és a MorphoLogic Kft. három éves munkájaként jött létre. Több mint 42 000 synset, melyből 2 000 synset az üzleti nyelvből, 650 synset a jogi nyelvből származik. Alapjául a Princeton WordNet 2.0 szolgált, melyből a *BalkaNettel* (Tufis, Cristea és Stamou 2004) közös fogalomhalmazokat kiválasztották és lefordították magyarra.

Az erőforrás tartalmaz kapcsolatokat a Princeton WordNettel és a MetaMorpho néhány igei vonzatkeretével is. Úgy gondolom, hogy ezen kapcsolatok felhasználásával – mivel a Princeton WordNet a VerbIndex-szel össze van kötve – segíthető a MetaMorpho és a VerbIndex összekapcsolása.

4.2.2. Szabadon elérhető magyar igei adatbázisok

Az alább ismertetésre kerülő adatbázisok korpuszokból statisztikai módszerekkel készültek. Közös jellemzőjük, hogy a korabeli nyelvtechnológiai szerelőszalag működésének hatékonyságát tükrözik. A fő felhasználási módjuk az elméleti nyelvészet területén az egyes igék és vonzatkereteik egymáshoz képesti gyakoriságának vizsgálata, de a későbbiekben szeretném őket felhasználni az AnaGramma elemzőrendszerben, valamint az ahhoz készülő modulokban is.

A *Magyar igei szerkezetek – A leggyakoribb vonzatok és szókapcsolatok szótára* című mű (Sass et al. 2010) a Tinta Kiadó gondozásában jelent meg. A szótár több társszerző által lektoráltan tartalmazza a leggyakoribb magyar nyelvű igei szerkezeteket, melyek automatikus előállítása Sass Bálint PhD disszertációján (Sass 2011) alapul, aki az igei vonzatkeretekkel és korpuszból való kinyerésükkel foglalkozott. A lektorált változat nem sokkal ezelőttig nem volt

DOI:10.15774/PPKE.ITK.2018.007

elérhető elektronikus úton, így számítógépes kutatásokhoz nem lehetett felhasználni, csak az alapjául szolgáló lektorálatlan, 28 millió szintaktikailag elemzett mondatból és félmillió igei szerkezetből álló erőforrást (Sass 2015). Ez utóbbi szolgál a *Mazsola* (Sass 2009) névre keresztelt eszköz online felületének[1] alapjául, melyből „kimazsolázható" az egyes igék kereteinek eloszlása illetve a különböző vonzatokhoz tartozó ige tövek is.

A *Mazsola* rendszer adatbázisa a 180 millió szót tartalmazó Magyar Nemzeti Szövegtárból (Váradi 2002) épült fel – mely a mai viszonylatban kicsinek számít. A Mazsola adatbázisa 18,3 millió olyan finit igés tagmondatot tartalmaz, amelyben az igék és a főnévi csoportok fejei, melyek argumentumai vagy módosítói az igének, annotálva vannak. A Mazsola elve a következő: a szintaktikailag megelemzett korpuszt tagmondatokra vágjuk, és a tagmondatokon belül megkeressük a mondatfában az ige alá tartozó argumentumokat, melyeket szótövük és nyelvtani esetük alapján (beleértve a névutókat is) megkülönböztetünk. Az így létrejött n-eseket gyakoriság szerint rendezzük, és egy algoritmussal kiválogatjuk azokat a gyakori lexikális elemeket is tartalmazó kereteket, melyek az őket tartalmazó keretekhez képest sokszor előfordulnak. Az így kiválasztott, megtartandó lexikális elemek gyakoriságát kivonjuk az absztraktabb szülő keretből, így megkapva annak tényleges gyakoriságát. Ezek után Sass Bálint eldobta a nagyon ritka kereteket, és csak a bizonyos küszöbértéknél[2] gyakoribbak találhatók meg az adatbázisban. Látható, hogy a módszer felépítése soros, és nagyban támaszkodik arra, hogy a szintaktikai elemzés, valamint a tagmondatokra bontás helyes volt, de az eljárás célja nem a fedés, hanem sokkal inkább a pontosság. Kiemelkedő érdeme az erőforrásnak, hogy sokáig az egyetlen szabadon elérhető magyar nyelvű statisztikai alapú igei vonzatkeret-adatbázis volt.

A *Tádé* egy az MNSZ-nél sokkal nagyobb korpuszon, az 589 millió szavas *Webkorpuszon* (Halácsy, Kornai, László et al. 2004), modern klaszterezéssel készült erőforrás (Kornai, Nemeskey és Recski 2016). Létrehozásának célja az opcionális

[1] `http://corpus.nytud.hu/mazsola/`

[2] Ez a küszöbérték a lektorálatlan félmillió igei szerkezet esetén 5.

DOI:10.15774/PPKE.ITK.2018.007

igei vonzatok megtalálása. A Tádé sokkal több potenciális vonzatkeretet tartalmaz, az infinitívuszt vonzó igék vonzatkereteit is ideértve, melyek a Mazsolából hiányoznak. Készítése során az 50-nél ritkábban előforduló kereteket kiszűrték, mégis szemmel láthatóan a pontossága sokkal alacsonyabb, mint a Mazsolának. Készítésekor inkább az F-mértékre optimalizáltak, mely magával vonta a pontossággal szemben magasabb fedésre való törekvést. Sajnos nem lett összehasonlítva a Mazsolával abból a szempontból, hogy hány közös és hány eltérő vonzatkeretük van. Így ha valaki pusztán az igei vonzatkeretekre kíváncsi, akkor előnyben részesíti a kisebb fedésű, de sokkal nagyobb pontosságú Mazsolát a Tádéval szemben.

A *Manócska* egy olyan igei-vonzatkeret erőforrás szerepét célozza meg, amely összehangolja és integrálja a meglévő, szabadon elérhető erőforrásokat[1] (Indig, Vadász és Kalivoda 2017). Gépi úton harmonizálva tartalmazza az eddig csak nyomtatott formában elérhető igei szótárat, a Mazsolából származó és a szótár alapjául szolgáló félmillió igei vonzatkeretet, a Tádét, a MetaMorpho magyar–angol változatának magyar oldalát, valamint a Kalivoda Ágnes mesterszakos szakdolgozatának mellékleteként szereplő adatokat, egy 27 083 igekötős igét felsoroló, kézzel ellenőrzött gyakorisági listát (Kalivoda 2016). Kalivoda Ágnes külön a Manócskához készített továbbá egy listát az infinitívuszt is vonzó igékről a *Magyar Nemzeti Szövegtár 2.0.4* alapján, mely külön is elérhető[2]. Az integrált erőforrások közös hiányossága[3], hogy nem rendelkeznek szemantikai információval, pusztán csak a felszíni jegyek alapján különböztetik meg a vonzatkereteket. Ezáltal a Manócska a ma elérhető legbővebb[4], nyílt hozzáférésű magyar igei vonzatkeret-adatbázis, mely az elődeit kiegészíti a *Linked Data* nyelvtechnológiai erőforrásokra értendő koncepciójával.

[1] `https://github.com/ppke-nlpg/manocska`

[2] `https://github.com/kagnes/infinitival_constructions`

[3] Itt eltekintek a későbbiekben bemutatásra kerülő *MetaMorphotól*, – mely részlegesen integrálásra került a *Manócskába*, – mivel az nyelvész intuíció alapján jött létre, nem pedig közvetlen korpuszstatisztikákból, így a benne szereplő keretek és szemantikai jegyeik nem feltétlenül tükrözik a statisztikát.

[4] Minden ma létező magyar nyelvű igei erőforrást közös keretbe integrálva tartalmaz.

DOI:10.15774/PPKE.ITK.2018.007

4.3. Az összekapcsolandó adatbázisok

4.3.1. MetaMorpho

A MetaMorpho egy tisztán szabályalapú, kereskedelmi gépi fordító rendszer (Prószéky és Tihanyi 2002), melynek a dolgozatban a magyar-angol változatát fogom ismertetni és használni. A rendszer különlegessége, hogy több mint 34 000, kézzel készült, igei vonzatkereteket leíró szabályával (melyek 17 000 magyar igét fednek le) máig a legnagyobb fedésű ilyen jellegű erőforrás magyar nyelvre. A rendszerben 27 bináris tulajdonság van, ami a szemantikus osztályokat reprezentálja, valamint 54 további morfológiai és más nyelvtani jellemző, melyek megszorításokat írnak le az argumentumokra nézve. A működése főbb vonalakban: a fordítórendszer mély szintaktikai elemzéssel megelemzi a forrásoldalon található szöveget, és a benne lévő szabályokra próbálja meg illeszteni azt. Az illeszkedő szabályok felépítik a párhuzamos célnyelvi reprezentációt, ami végül a kimenet lesz.

Minden, az igei vonzatkeret azonosítását célzó szabály egy igét tartalmaz lexikális és morfológiai megszorításokkal, valamint az argumentumaira morfológiai, szófaji és szemantikai megszorításokat. Ezenfelül, ha szükséges, az argumentumok lexikális megszorításokat is tartalmaznak. Néhány argumentum opcionális, azaz nem szükséges feltétlenül realizálódnia a mondatban ahhoz, hogy a szabály megfeleljen. A (16) példán láthatjuk, hogyan néz ki egy tipikus MetaMorpho szabály és egy erre illeszkedő mondat.

(16)
```
HU.VP = SUBJ(human=YES)  + TV(lex="ábrándozik") +
EN.VP = SUBJ             + TV(lex="dream")      +
```
[Minden tudós]$_{\texttt{SUBJ}}$ [ábrándozik]$_{\texttt{TV}}$
```
COMPL#1(pos=N, case=DEL)
COMPL#1(prep="about")
```
[a tudományos áttörésről]$_{\texttt{COMPL\#1}}$

Mivel az adatbázis egy gépi fordítórendszerből származik, minden magyar szabályhoz tartozik egy angol megfelelő, mely tartalmazza a magyar elemzésnek megfelelő argumentumokat és azokat az elemeket, amik szükség szerint új tokent hoznak létre, hogy a magyarnak szemantikailag megfelelő angol keretet tudják

DOI:10.15774/PPKE.ITK.2018.007

alkotni. Például ilyen a delatívusz a (16) példában, mely az angol oldalon az *about* prepozícióval realizálódik.

vonzatkeret típus	előfordulások száma	%
SUBJ TV OBJ	5 535 334	30,22%
SUBJ TV COMPL#1	4 501 736	24,57%
SUBJ TV OBJ COMPL#1	3 859 952	21,07%
SUBJ TV	2 465 005	13,46%
(13 más tipus)	1 957 700	10,68%
összesen:	18 319 727	

4.1. táblázat. Igei vonzatkeret előfordulások a Magyar Nemzeti Szövegtárban.

A rendszerben minden szabály egyenrangú, ezért a szabályok egy lapos listaként vannak tárolva, melyben az adott mondat a megfelelő szabályra illeszkedik. Ahhoz, hogy meghatározhassam a szabályok valóélet-beli előfordulásának gyakoriságait, felhasználtam a Mazsola adatbázist (lásd a 4.2.2. fejezet). Leképeztem a Mazsola grammatikai eseteit a MetaMorpho igei vonzatkeret terminológiájára: alanyeset=SUBJ, tárgyeset=OBJ, a többi eset és névutó=COMPL*. Ezeket a címkéket felhasználva megszámoltam az előfordulásokat a korpuszban található minden igei vonzatkerethez. A 4.1. táblázatban látható, hogy a leggyakoribb 4 típus az összes igei vonzatkeret 88%-át teszi ki a korpuszban. Ez alapján csak az *intranzitív*, *mono-tranzitív* (tárgy vagy egyéb nem tárgyesetű argumentum) és a *ditranzitív* (tárgy és még egy argumentum) kereteket vettem számításba a későbbi lépésekben, hogy jó korpuszfedésen, mégis kevés típussajátosságtól akadályoztatva dönthessek az erőforrások összekötésének hatékonyságáról.

4.3.2. VerbIndex

A *VerbIndex* egy több külön erőforrásból létrehozott igei lexikon, amely a SemLink Projekt része (Loper, Yi és Palmer 2007). Fontosabb alkotórészei a *VerbNet*, mely az angol igéket sorolta a Levin-féle osztályokat kiterjesztve hierarchikusan egymásba ágyazott osztályokba aszerint, hogy milyen vonzatkereteik vannak[1] és a

[1] Tehát látható, hogy nemcsak az egyes predikátumok szerint történik az osztályozás, hanem az őket tartalmazó igék szerint is.

DOI:10.15774/PPKE.ITK.2018.007

Prop Bank, mely a korpuszokban található atomi állítások szemantikai viszonyai alapján osztályozást hozott létre, gyakorlatilag az igei vonzatkeretek között. A VerbIndex az argumentumokra vonatkozó szintaktikai és szemantikai megszorításokat tartalmaz, továbbá az argumentumok tematikus szerepei is meg vannak adva. Az igék meg vannak különböztetve a Princeton WordNet-beli (lásd a 4.2.1.1. fejezet) jelentésük alapján is. Így kizárható az azonos alakú, de több jelentésű igék vonzatkereteinek összekeveredése. A Prop Bank-ből származó szemantikus reprezentáció tartalmazza többek között az argumentumokhoz tartozó *tematikus szerepeket*, melyeket jól lehetne hasznosítani abban az esetben, ha magyar nyelvre át lehetne vinni ezeket.

Például a MetaMorpho rendszerből származó *ábrándozik* vonzatkeretnek az angol megfelelője a VerbIndexben a *dream* ige megfelelő kerete, mely a `wish-62` osztályba tartozik és a (17) példában látható leírás – amiben a *dream* ige mindenhol behelyettesíthető – található róla az erőforrásban.

(17) *I* *wished* *it.*
`NP` `V` `NP`
`Experiencer` `V` `Theme<-sentential>`
desire(E, `Experiencer`, `Theme`)

4.4. Az igei vonzatkeretek adatbázisainak összekapcsolása

Az összekapcsolás ideális kimenete tehát a (18) példában látható eredményt adná az *ábrándozik* esetében, melyben a magyar nyelvre is elérhetővé válnak a szemantikai annotációk (az erőforrások kapcsolatait lásd az E.1. ábrán). A tényleges összekapcsolás előtt azonban előzetes vizsgálatokat végeztem, hogy a leszűkített igei osztály esetében (lásd a 4.3.1. fejezet) milyen egyértelműsítési feladatok merülhetnek fel, melyek a gép számára megnehezítik az összekapcsolást.

DOI:10.15774/PPKE.ITK.2018.007

(18)
```
HU.VP =  SUBJ(human=YES) + TV(lex="ábrándozik") +
EN.VP =  SUBJ             + TV(lex="dream")      +
wish-62: Experiencer        V
  COMPL#1(pos=N, case=DEL)
  COMPL#1(prep="about")
  Theme<-sentential>
```
desire(E, Experiencer, Theme)

4.4.1. Előzetes vizsgálatok

A WordNet úgy tud részt venni a vonzatkeretek összekapcsolásában, hogy a különböző nyelvek igéit jelentés szerint megfelelteti egymásnak, ezzel egyértelműsítve a homonimákat. Ezért megnéztem (lásd a 4.2. táblázat), hogy a WordNetben található igék hogyan viszonyulnak a VerbIndex-beli igékhez, és mennyi közülük a nehezen kezelhető *többszavas ige (phrasal verb)*. Ennek oka, hogy a többszavas lexémák különbözőképpen vannak reprezentálva a magyar és angol erőforrások között – de még a független angol nyelvű erőforrások között is –, ami problémát jelenthet, amikor az összekötésre jelölt párokat vizsgáljuk a két erőforrásban. Szerencsére az ilyen igék kevesebb mint fele lett kidolgozva a VerbIndexben, így ebbe a problémába nemigen ütközhetünk a későbbiekben.

	igék száma
igék száma a WordNetben	7 440
többszavas igék száma a WordNetben	1 410
többszavas igék száma a VerbIndexben	404
többszavas igékből származó igetövek száma a VerbIndexben	223

4.2. táblázat. Többszavas igék a WordNetben és a VerbIndexben.

A VerbIndex igéinek vizsgálata után (lásd a 4.3. táblázat) arra jutottam, hogy a MetaMorpho rendszernél nem sokkal bővebb a tényleges, kerettel rendelkező angol igekészlet – az összekapcsolt erőforrások eltérő kidolgozottsági szintjei miatt –, viszont az igék majdnem kétharmada egyértelműen, csak egyszer szerepel, ami megkönnyíti a későbbi összekapcsolást.

DOI:10.15774/PPKE.ITK.2018.007

	igék száma
igék a VerbIndexben	6 343
keret nélküliek, csak említés szintjén szerepelnek	2 057
kerettel rendelkezők, összekapcsolhatók	4 286
csak egy osztályban szereplő igék	2 957

4.3. táblázat. Az igék tulajdonságai a VerbIndexben.

A MetaMorpho rendszer vizsgálata után a benne található igék és a szabályok arányainak összehasonlítása alapján (melyet a 4.4. táblázat mutat) úgy találtam, hogy 10:1 arányban vannak az igei vonzatkeretek és az angol igék. Ennek két oka van. Egyfelől az, hogy a MetaMorpho rendszerben külön szabályokat kaptak a változatos idiomatikus konstrukciók, amik az igék többségénél a szabályok nagy részét adják. Ez az összes szabály kicsivel több mint egyharmadát érinti. Másfelől az, hogy a MetaMorpho fejlesztése során nem volt cél az angol oldalon jó fedést elérni, mert a fordítások szempontjából elég volt a magyar oldali jó fedés és az angol fordítás pontosságának együttes fennállása. A vizsgálatok azt is kimutatták, hogy a MetaMorpho igéinek 42%-a a VerbIndex több osztályában is szerepel. Ebből következően az egyes MetaMorpho szabályokhoz tartozó jó osztályt is ki kell választani az összekapcsolás során, nem csak a megfelelő keretet. A későbbiekben ezt a két összekapcsolási szintet különböztetem meg.

	darab
igei vonzatkeretek	30 292
egyedi angol igető	3 505
angol igetövek, amik nincsenek a VerbIndexben	920
angol igetövek, amik a helyesírás-ellenőrző számára ismeretlenek	143
idiomatikus vagy lexikálisan megszorított angol keretek	10 694
idiomatikus vagy lexikálisan megszorított magyar keretek	8 347

4.4. táblázat. Az igék tulajdonságai a MetaMorpho rendszerben.

4.4.2. Az összekapcsolás módja

A módszerem abból áll, hogy minden lehetséges kombinációt végigpróbálva, a keretek között különböző egymásra épülő szűrők segítségével próbálom kiszűrni

DOI:10.15774/PPKE.ITK.2018.007

azokat a kapcsolatokat, amelyek valamilyen szempontból hamisnak bizonyulnak. Minden szűrő után megvizsgáltam a megtartott és kiszűrt elemeket, majd az eredménynek megfelelően javítottam a szűrőn vagy definiáltam egy újabb szűrőt, melyet sorba kapcsoltam a már meglevőkkel. A szűrő nélküli *maximális összekapcsolást* tekintettem az alapvonalnak. Ez csak az igék egyezését vizsgálta, kiszűrve a kereteket, amelyek csak az egyik erőforrásban található igéket tartalmazzák.

Mindkét erőforrás kézzel készült, ezért a felbontásuk nagyon nagy határok között mozog, egyes részeik nagyon jól ki vannak dolgozva, mások kevésbé. Az összekapcsolás első fázisában az egyes MetaMorpho szabályok a VerbIndex osztályokkal az egyező igék megtalálása által lettek összekapcsolva, melyek száma és aránya segít eldönteni, hogy mennyire kompatibilis a két erőforrás, és érdemes-e további munkát fektetni az összekapcsolásba. A második szakaszban pedig az egyes argumentumok száma és megszorításai alapján a logikailag lehetetlen összekapcsolásokat zártam ki.

Több külső erőforrást is felhasználtam, úgy mint a WordNeteken keresztüli kapcsolatokat és az általam[1] készített ontológiákat, melyekről a 4.4.5. fejezetben lesz bővebben szó. A bevont erőforrások az összekapcsolás pontosságát célozták növelni szintaktikai és szemantikai szempontból és egyúttal a rossz kapcsolatokat minél pontosabban kiszűrni.

A fentiek alapján egy adott MetaMorpho–VerbIndex összekapcsolásban a specifikus MetaMorpho szabályok kapcsolatai a következő 5 típusba sorolhatóak be:

(i) Nincs lehetséges VerbIndex kapcsolat.

(ii) Egyértelmű (egy-egy) leképezés: csak egy kapcsolat van, ami lehet

  (iia) helyes

  (iib) helytelen

(iii) Többértelmű (egy-a-többhöz) leképezés: több mint egy kapcsolat van, amik

  (iiia) tartalmazzák a helyes leképezést (ha létezik ilyen) vagy

  (iiib) nem tartalmazzák azt (ami lehet azért, mert nem is létezik).

[1]Köszönöm Simonyi Andrásnak a segítséget!

DOI:10.15774/PPKE.ITK.2018.007

A két erőforrás különböző felbontása és fedése miatt a maximális leképezés sok (iib) és (iiib) típusba tartozó, nem megfelelő leképezést tartalmaz. Pontosabban sok olyan keretet, amely csak az egyik erőforrásban található meg, annak ellenére, hogy a bennük levő ige magában mindkét erőforrásban megtalálható. A célom a szűrők hozzáadásával az ilyen esetek kiszűrése volt.

4.4.3. A két erőforrás különbségei

A két erőforrás közötti alapvető különbséget az okozza, hogy míg a MetaMorpho egy lapos lista formában tartalmazza az egyes szabályokat, szükség szerint az opcionális argumentumokat feltüntetve, addig a VerbIndex a szemantikai szempontból azonosan működő igéket egy osztályba sorolja, és az osztályhoz sorakoztatja fel az egyes lehetséges vonzatkereteket, nem jelölve az opcionális argumentumokat. Az osztályok egymáshoz képest hierarchiában is lehetnek, ami azt jelenti, hogy az alosztályok öröklik a szülőosztály tulajdonságait, és tovább finomítják őket. Tehát elő kellett állítani mindkét oldalon egy minden származtatott tulajdonságot is tartalmazó bejegyzést az összehasonlításhoz.

A VerbIndex oldalán szükség volt egy egyszerű konverzióra az argumentumok reprezentációja miatt, mivel a MetaMorpho rendszerben a prepozíciók az argumentumok egy tulajdonságaként vannak reprezentálva, míg a VerbIndex prepozíciói külön elemként szerepelnek a keretben. Az eljárás során a VerbIndex prepozícióit az argumentumok tulajdonságaivá konvertáltam. Továbbá, míg a VerbIndex a prepozíció osztályok kombinációival operál, a MetaMorpho mindig megjelöli a konkrét prepozíciót. Ennek a kezelésére egyszerűen azt vizsgáltam, hogy a MetaMorpho által megjelölt prepozíció benne van-e a VerbIndex által megjelölt halmazban.

4.4.4. A szűrők

A maximális összekapcsoláshoz képest kiszűrtem azokat a szabályokat, ahol az argumentumok átrendezése volt szükséges, illetve azokat, ahol a MetaMorpho rendszerben nem egyezik meg az argumentumok száma a szabályok angol és magyar oldalán. Továbbá egy harmadik szűrővel az összekapcsolások közül kiszűrtem azokat, amelyeknél a MetaMorpho argumentumszáma és a VerbIndex keret

DOI:10.15774/PPKE.ITK.2018.007

argumentumszáma nem egyezett. Ezenkívül az opcionális argumentumokat tartalmazó szabályokat is kiszűrtem, mivel a VerbIndex oldaláról több szabálynak is megfelelhettek volna. Nehézkes kezelhetőségük miatt azokat a szabályokat is kirostáltam, amelyekben az egyes argumentumok lexikális megszorítást tartalmaznak. Így biztosítottam, hogy az argumentumoknál minden esetben 1:1 leképezés jöhessen létre. Az így kizárt szabályok nagy része átalakítható lenne úgy, hogy megfeleljenek a kritériumoknak. A kísérleteimben a fő elképzelésre koncentrálva az egyszerűen összeköthető szabályokra redukáltam a vizsgált elemek körét.

További megszorításként értelmeztem azt a sajnos csak kevés esetben megtalálható kapcsolatot, ahol a MetaMorpho szabályok igéi össze vannak kötve a Magyar WordNet egy synsetjével, mert ha a hipotetikus kapcsolat a két erőforrás adott vonzatkeretei között helyes, akkor a Magyar és a Princeton WordNet összekötésén keresztül a VerbIndex Princeton WordNettel összekötött igéi elérhetőek kell, hogy legyenek. Ez az út sokszor hiányos, vagy pontatlan volt, ezért kevés esetben sikerült ténylegesen alkalmazni.

Az argumentumok prepozícióin értelmezett megszorítások újabb lehetőséget kínáltak a helytelen összekötések kiszűrésére, mivel a két erőforrásban jelölt prepozíciós megszorításoknak is kompatibilisnek kellett lenniük egymással. Az utolsó két szűrő az összekapcsolt elemek közötti szintaktikai és szemantikai megszorítások kompatibilitását igényelte. A megszorítások kezelésének részleteit a következő fejezeteben részletesen tárgyalom.

4.4.5. A megszorítások ontológiái

A megszorításokat leíró formalizmusok különösen a szemantikai megszorítások esetén különböznek a két erőforrás között. A különbségek feloldása érdekében be kellett vezetni egy explicit formális reprezentációt az egyes kapcsolatok kezelésére két manuálisan készített OWL ontológia formájában. Azért esett az ontológiára a választás, mert mindkét oldalon egymáshoz képest hierarchikusan épülnek fel a megszorítások jellemzői.

DOI:10.15774/PPKE.ITK.2018.007

4.4.5.1. A szintaktikai megszorítások ontológiája

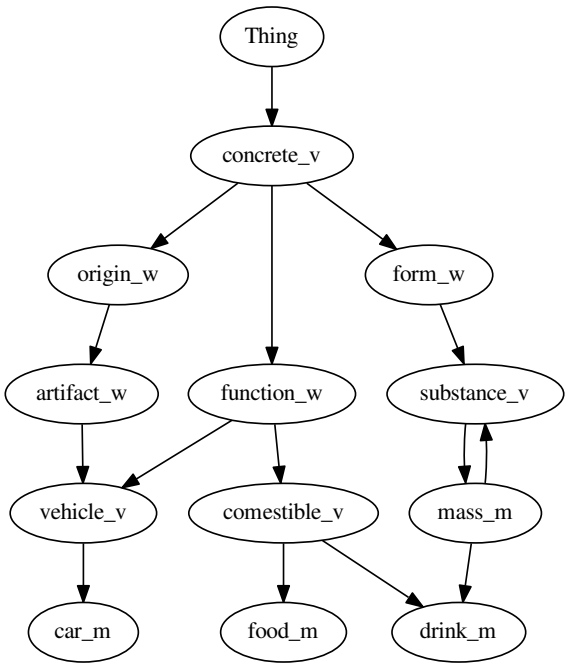
A VerbIndex-szel ellentétben, ami 40 bináris jellemzővel írja le a szintaktikai megszorításokat, a MetaMorpho leírása az angol keretek tekintetében csak 4 többértékű attribútumot alkalmaz: *mellékmondat típusa (clausetype)* 6 lehetséges értékkel, *igeidő (tense)* 3 lehetséges értékkel, és a következő bináris jellemzőket: *birtokos szerkezet (poss)* és *szám (num)*. A szintaktikai ontológiám tartalmazza az összes VerbIndex jellemzőt és az összes MetaMorpho attribútum-érték párt OWL osztályok formájában, valamint a logikai kapcsolataikat ekvivalencia axiómákkal reprezentálva. Például a `genitive`≡`poss_YES` axióma azt a tényt fejezi ki, hogy a VerbIndex birtokosságot kifejező *(genitive)* jellemzője megfelel a MetaMorpho rendszerbeli *poss*:*YES* hasonló jelentésű attribútum-érték párnak. Annak ellenére, hogy az axiómák nagy számban egyszerű ekvivalenciákat tartalmaznak, néhány axióma elég komplex. A VerbIndex *sentential* jellemzője például csak a MetaMorpho 7 különböző attribútum-érték párjának diszjunkciójából állítható elő.

4.4.5.2. A szemantikai megszorítások ontológiája

Mind a VerbIndex, mind a MetaMorpho kevés számú szemantikus kategória kombinációival írják le a megszorításokat, mely csoportok ontológiába vannak szervezve. Ennek ellenére a két ontológia eléggé különböző, és mindkettő tartalmaz olyan fogalomköröket, amelyeket nehéz megfeleltetni a másik ontológiában találhatóaknak. Például ilyen a MetaMorpho *punct* és a VerbIndex *communication* csoportja. A két erőforrás hasonló osztályokat meglepően különbözőképpen értelmez: a MetaMorpho rendszerben az események lehetnek az *abstract* fogalomkörben, míg a VerbIndexben az *event* és az *abstract* kategóriáknak nincs közös metszete.

Ezen különbségek tükrében úgy döntöttem, hogy a két rendszer szemantikai kategóriái közötti logikai kapcsolatokat egy egységes, kézzel készített megszorítási ontológiában reprezentálom, amely tartalmazza mind a két rendszer eredeti ontológiáját, valamint a szükséges áthidaló fogalmakat és axiómákat (lásd a 4.1. ábra). Az áthidaló fogalmak magas szintű fogalmak, amelyek a 4.2.1.2. fejezetben említett EuroWordnet felső ontológiájából származnak. Ezek olyan szervező

DOI:10.15774/PPKE.ITK.2018.007



4.1. ábra. A szemantikai megszorítások ontológiája egy része. (Az egyes osztályok forrása fel van tüntetve azok végződésében: v=VerbIndex, m=MetaMorpho, w=EuroWordNet felső ontológia.)

eszközök, amelyek segítenek kifejezni a logikai kapcsolatokat a két erőforrás kategóriái között tömör és fogalmilag tisztázott módon. Például mindkét ontológia tartalmaz néhány funkcionális osztályt, úgy mint *drink* (MetaMorho) vagy *vehicle* (VerbIndex), de egyikük sem tartalmazott az EuroWordNetéhez hasonló általános *function* elemet. Ezen osztály hozzáadása az ontológiához segítette az általánosítások kifejezését a funkcionális kategóriák körében (mivel a fentiek mind a VerbIndex *concrete* kategóriájának alkategóriái).

Mivel sem a MetaMorpho, sem a VerbIndex szemantikai megszorításainak ontológiája nincs részletesen dokumentálva[1] az egyes kategóriák értelmezésének szempontjából, sok kategória esetében a használatuk részletes elemzése után in-

[1] A MetaMorpho magyar oldalának vázlatos leírása ugyan elérhető (Gábor et al. 2008), de a pontos részletek az üzleti titok tárgyát képezik.

DOI:10.15774/PPKE.ITK.2018.007

tuícióm szerinti új, áthidaló axiómákat adtam az ontológiához. Az így létrehozott OWL ontológiámban (lásd a 4.1. ábra) az egyes fogalmak forrását (VerbIndex, MetaMorpho vagy EuroWordNet) a végződésükben jelöltem. Tartalmazza a két erőforrás összes kategóriáját (29 és 47 osztályt), valamint 15 osztályt az EuroWordNet felső ontológiájából. Nincsenek külön, névvel ellátott tulajdonságok vagy individuumok, és a 129 axióma mindegyike a `subClassOf`, `equivalentClass` vagy `disjointWith` kapcsolatok valamelyikének fennállását mondja ki különböző osztályok Boole-kombinációi között.

A reasoner

A két bemutatott megszorítási ontológia a két erőforrás vonzatkeretei közötti kompatibilitás vizsgálatát egy *reasoning* problémává redukálta: a két megszorítás akkor és csak akkor kompatibilis, ha a két ontológiából nem következik, hogy a megfelelő (elég komplex) ontológiai osztályok diszjunktak. Az általános megoldás szükségessé tette egy *reasoner* szoftver bevezetését. Mivel a két ontológia csak Boole-kombinációkat tartalmazó axiómákból áll, egy egyszerű propozíciós reasoner elég lett volna, de végül a kiforrott és kiváló OWL támogatást adó *Racer OWL reasonert* (Haarslev et al. 2012) alkalmaztam, ami az OWLink kliens-szerver protokollon keresztül érhető el (Liebig et al. 2011).

4.4.6. Az információátvitel sikerességének kiértékelése

A rendszer pontosságának vizsgálata egy véletlenszerűen válogatott 400 keretes mintával történt, mely a MetaMorpho rendszerből a sikeresen (egyértelműen vagy többértelműen) leképezett elemekből származott. A mintán a helyes összekötések két független annotátor döntéseinek egy harmadik általi összefésülésével lettek meghatározva[1]. A vizsgált halmaz tartalmazott 90 vonzatkeretet, mely nem volt összeköthető egy VerbIndex bejegyzéssel sem, ezért ezeket eltávolítottam a listából. A maradék vonzatkeretek alkották a gold sztenderdet.

Mivel a referenciaadat nem reprezentálja jól a teljes MetaMorpho adatbázist, csak azokat a kereteket vizsgáltam, amelyek benne voltak a mintában, ezért csak

[1]Mindkét annotátor csak a rendszerek ismeretéhez mérten vizsgálta az összekapcsolások jóságát, melyben véleményük között alig volt eltérés. Ennek ellenére mindketten elégedetlenek voltak a vizsgált rendszerek néha önkényes döntéseivel.

DOI:10.15774/PPKE.ITK.2018.007

az összekötés pontosságát lehet biztonságosan megállapítani. Minden szűrő kimenetét megvizsgáltam a következő módon: amennyiben egy MetaMorpho bejegyzés egyértelműen lett leképezve, és a leképezés megegyezett a gold sztenderd-bélivel, akkor helyesnek számított, egyéb esetben pedig helytelennek. Ha viszont a keret nem egyértelműen lett leképezve, akkor halmaztartalmazást vizsgáltam az egyenlőség helyett: amennyiben a helyes VerbIndex bejegyzés benne volt a gép által adott halmazban, akkor a leképezés helyesnek számított, egyéb esetben pedig helytelennek.

	összekapcsolt elemek száma	
	egyértelmű	**többértelmű**
maximális összekapcsolás (alapvonal)	431	26 560
argumentumátrendezés szükséges lehet	291	12 664
az argumentumszám nem egyezik a MetaMorpho angol és magyar oldala között	285	12 347
mono- és ditranzitív konstrukciók	267	10 146
az argumentumszám nem egyezik a MetaMorpho és a VerbIndex között	2 301	7 745
WordNet leképezésnek megfelel	2 181	6 858
prepozíciós megszorításoknak megfelel	2 929	4 610
ontológia (szemantikai megszorítások)	2 967	4 455
ontológia (mindkettő)	2 733	3 286

4.5. táblázat. Az összekapcsolt elemek száma az egymásutáni szűrők alkalmazása után.

A végső leképezés négyszer több egyértelmű leképezést adott, mint az alapvonalnak számító leképezés, valamint a többértelmű leképezések száma radikálisan csökkent. Az összekötés eredményét a teljes halmazon a 4.5. táblázat mutatja. A szűrők pontossága a 4.6. táblázatban látható. A leképezés – figyelembe véve a felhasznált erőforrások korlátait – majdnem tökéletes a gold sztenderdben szereplő és a gép által egyértelműen összekötött MetaMorpho keretek esetében.

4.4.7. A harmonizáció problémái

Számtalan dolog nehezítette meg a MetaMorpho és a VerbIndex kereteinek összekötését. Néhány közülük az erőforrásokból fakadó probléma volt.

DOI:10.15774/PPKE.ITK.2018.007

	összekapcsolt elemek száma			
	egyértelmű		többértelmű	
maximális összekapcsolás (alapvonal)	100%	(9)	98,38%	(183)
argumentumátrendezés szükséges lehet	100%	(9)	98,38%	(183)
az argumentumszám nem egyezik a MetaMorpho angol és magyar oldala között	100%	(9)	98,38%	(183)
mono- és ditranzitív konstrukciók	100%	(9)	98,38%	(183)
az argumentumszám nem egyezik a MetaMorpho és a VerbIndex között	100%	(114)	96,29%	(78)
WordNet leképezésnek megfelel	100%	(101)	97,14%	(68)
prepozíciós megszorításoknak megfelel	90,43%	(104)	79,62%	(43)
ontológia (szemantikai megszorítások)	90,98%	(111)	76,59%	(36)
ontológia (mindettő)	92,59%	(100)	70,83%	(17)

4.6. táblázat. Az összekapcsolás pontossága és az összekapcsolt elemek száma a kézzel annotált gold sztenderd minta alapján.

Először is, a MetaMorpho igei vonzatkeret-adatbázisa nem általános használatra szánt nyelvtechnológiai erőforrásnak készült, sokkal inkább egy specifikus gépi fordítórendszer részének. Ennek következtében az angol oldal lexikális fedése nem kellően nagy, amit körülírásszerű fordításokkal próbáltak a szerzők kompenzálni, melyek egy VerbIndex-szerű lexikális erőforrásban nehezen kereshetőek. A MetaMorpho angol igei keretei nagy számú idiómát és félig kompozícionális elemet tartalmaznak – melyeknél egy vagy több argumentum lexikálisan kötött, például *take part in sg.* ‘részt vesz valamiben’, *make room for sg.* ‘helyet csinál valaminek’ –, amelyek a VerbIndexből teljes egészében hiányoznak.

Továbbá a jellemzők, amelyek leírják a szemantikai megszorításokat, a magyar oldalon jól működnek egy gépi fordítórendszer esetében, de a szigorúan formális definíciók hiánya nehézségeket okoz, ha le akarjuk képezni őket egy másik jellemzőrendszerre.

Másodsorban, a VerbIndexnek rekurzív, komplex szemantikai megszorításai vannak (lásd a 4.4.5. fejezet), amiket nehéz feldolgozni. Annak ellenére, hogy a VerbIndex egy jól kidolgozott erőforrás, a szemantikai jellemzők és kategóriák, melyeket a szintaktikai keretleírásokban használ, nincsenek jól dokumentálva, vagy homályosan dokumentált erőforrásokból származnak, amely megnehezíti az értelmezést.

DOI:10.15774/PPKE.ITK.2018.007

Úgy találtam, hogy a VerbIndex néha hiányos. Például a *kopog* (*knock* `sound_emission-43.2`) ige intranzitív formájában csak a *Téma* tematikus szerep található meg, habár úgy gondolom, hogy a *Valaki kopogott. (Somebody knocked.)* mondat esetében az alany *Ágens*.

Végül a WordNetnek megvannak a saját problémái. A főnévi hiperníma hierarchiája, ami taxonómiai hálózatként hasznos, ugyanakkor nem reprezentálja az általános nyelvhasználat felbontását, mely sokszor durvább. És a különböző nyelvű és verziójú WordNetek különbségei szintén problémákat okoztak.

4.4.8. Egy mondatelemző-alapú megközelítés

A MetaMorpho szabályaihoz a szabályírók egyszerű példákat is megadnak[1], melyekkel fejlesztés közben tesztelték a rendszer működését. Az így megadott példamondatoknak pontosan arra az egy szabályra szabadott illeszkedniük, amelyhez meg vannak adva. Egy kísérletben ezeket a mondatokat felhasználva próbáltam egy mondatelemző segítségével a referenciaadatban található szabályokhoz meghatározni a megfelelő VerbIndex-beli keretet és az argumentumok tematikus szerepét. Ehhez a referenciaadatot manuálisan ki kellett bővíteni példamondatokhoz tartozó tematikus szerepekkel, (az egyszerűség kedvéért) az angol mondatnak megfelelő sorrendben. Ahol nem volt megfeleltethető egymásnak a két erőforrás, ott manuálisan pótoltuk a tematikus szerepeket. Az így előállt új 400 mondatos gold sztenderd adaton meg tudtuk mérni, hogyan teljesít a state-of-the-art angol nyelvű szemantikus elemző, mely eredményéből kinyerhetők a tematikus szerepek.

Először is lefordítottam a mondatokat a MetaMorpho segítségével angolra, ami azért volt fontos lépés, mert más fordítórendszer valószínűleg máshogy fordított volna bizonyos mondatokat, ami növelte volna a hiba esélyét, viszont így a szabályoknak megfelelő angol mondatokat kaptuk meg. A kapott angol mondatokon lefuttattuk az elemzőprogramot, amely felismerte a predikátumokat.

A PathLSTM (Roth és Lapata 2016), state-of-the-art elemzőre esett a választásunk, mely lexikalizált függőségi-út beágyazásokat és számos bináris jellemzőt

[1]Ezek a mondatok többnyire *János szereti Marit.* komplexitásúak voltak.

DOI:10.15774/PPKE.ITK.2018.007

használ a szemantikai elemzéshez. A tokenizáláshoz, függőségi elemzéshez és szemantikai predikátum azonosításhoz és egyértelműsítéshez a forráskód dokumentációjában ismertetett szerelőszalagot használtuk (Roth és Lapata 2017), amely a Stanford CoreNLP WSJ tokenizálójából (Christopher D Manning et al. 2014), a Bohnet függőségi elemzőből (Bohnet 2010), és a mate-tools predikátumkeret felismerőből (Björkelund, Hafdell és Nugues 2009) áll. A PathLSTM programot egy előre betanított modellel futtattuk, amely támogatta a ProbBank-féle predikátumszerep címkéket, és ezeket konvertáltuk a VerbIndex által ismert tematikus szerepekre a SemLink projekt ProbBank–VerbNet leképezésének felhasználásával (Loper, Yi és Palmer 2007)[1].

Csak a fő predikátumokat vettük figyelembe, amelyek egyeztek a finit igével, a többi azonosított predikátumot eldobtuk. Az azonosított predikátumok változatos módokon tartalmaztak hibákat. Volt, hogy az ige rosszul volt lemmatizálva és így nemlétező keretre hivatkozott, valamint az argumentumok felismerése függetlenül történt a keretektől, így ritkán kaptunk az adatbázisban megtalálható teljes keretet. Ezenfelül, mivel a ProbBank–VerbNet nem mindig adott egyértelmű és teljesen egyező eredményt, a következő egyértelműsítési szabályokat alkalmaztuk: Minden VerbNet keretet, amely megfelelt a SemLinkben az elemzett PropBank predikátumnak, de tartalmazott olyan argumentumot, amely nem szerepelt az elemzésben, *részleges egyezésnek* számoltuk, ellenkező esetben *teljes egyezésnek*. Ha volt teljes egyezés, akkor a részleges egyezéseket eldobtuk és a legnagyobb fedésű egyezést választottuk. Ez utóbbi eljárást végeztük el akkor is„ ha csak részleges egyezés volt. Azon részleges egyezéseket részesítettük előnyben, amelyek esetén a VerbNetben kevesebb argumentum volt, mint az elemzésben. A többi esetet csak ezek után vizsgáltuk. Ezen szabályokra alapozva a legjobban egyező VerbNet keretet és tematikus szerepet tudtuk minden mondathoz kiosztani.

Mondat- és címkealapú kiértékelést is végeztünk (Indig, Simonyi és Miháltz 2018) (lásd a 4.7. táblázat), melyben csak a pontosságot vizsgáltuk. Összesen 429 mondatot elemeztettünk le, de csak 327 mondat maradt, amiben legalább egy tematikus szerep maradt a keret konzisztencia ellenőrzése után. A referenciaadat

[1]Az egész elemzőrendszer az előre betanított modellel együtt innen elérhető: `https://github.com/microth/PathLSTM`.

DOI:10.15774/PPKE.ITK.2018.007

főképpen egyszerű kereteket tartalmazott, ahol egyszerűen át lehet fordítani az argumentumokat angolról magyarra, mivel nem volt szükség átrendezésre. Azon mondatoknál, ahol mégis szükség volt átrendezésre, az egyszerűség kedvéért úgy állapítottuk meg a referenciaadatban a tematikus szerepeket, hogy azok az angol sorrendnek feleljenek meg, így könnyen kiértékelhetőek legyenek[1].

	Jó	Összes	Pontosság (%)
Címkék száma	428	602	71.096
Keretek száma	193	327	59.021

4.7. táblázat. Az elemzőprogrammal történő tematikus szerep címkézési feladat eredménye.

Egy mondatelemző-alapú megoldástól jobb eredményeket vártunk, de azt láttuk, hogy a bonyolult statisztikai általánosítások nem működnek jól együtt a kézzel készült, nyelvészetileg motivált MetaMorphoval és VerbIndexel. Az elemzések nagyon inkonzisztensek voltak, és sok hibát ki lehetett volna javítani az elemzőn belül. Például néhány inflexiós ige a lemmatizálás hibájából olyan szótövet kapott, amelyből származtatott osztály nem létezett a ProbBankban. Sok esetben fordult elő, hogy a predikátum nem egyezett egy várt osztállyal sem, mivel argumentumok hiányoztak vagy fölösleges argumentumok voltak jelen. Véleményem szerint, ha egy ismert igét talál a rendszer, jobb lenne, ha a létező keretek mintáiból választana ahelyett, hogy megpróbálja általánosítani őket, mivel a további feldolgozás nagyban támaszkodik a keretek jóságára.

Ezen hibás működés miatt az eredmények sokkal rosszabbak lettek, mint amit jogosan elvárhattunk volna egy ilyen fejlett, statisztika-alapú mondatelemző módszertől. Ezért azt a következtetést lehet levonni, hogy a javasolt szabályalapú rendszer a nyelvek közötti tematikus szerepek átvitelének feladatában jobban teljesít, mint a leírt statisztikai mondatelemző-alapú módszer.

[1] A való életben egy szabályalapú rendszer esetén könnyű ezt az átrendezést elvégezni úgy, hogy a felismert tematikus szerepek sorrendje megegyezzen a magyar nyelvű argumentumokéval.

DOI:10.15774/PPKE.ITK.2018.007

4.5. Összefoglalás és kapcsolódó tézisek

A fejezetben bemutattam a *Linked Data* fogalmát. A módszer erőforrásokra vonatkoztatott változatának ismertetése után bemutattam néhány példát az összekapcsolt erőforrásokra. Majd ezen a vonalon elindulva, a bemutatásukat követően a kétnyelvű, magyar–angol MetaMorpho adatbázis és az angol VerbIndex összekapcsolását tűztem ki célul, hogy nyelvfüggetlen annotációt tudjak automatikusan átvinni a szemantikus információban gazdagabb VerbIndexből a MetaMorpho rendszerbe.

6. Tézis. *Létrehoztam egy automatikus módszert az 1-, 2- és 3-vonzatú igék magyar–angol vonzatkeretpárjainak összekapcsolására, melynek eredményeképpen sikerült angolról magyarra átvinni a megfelelő tematikus szerepeket.*

A tézist alátámasztó közlemények: [11, 12, 4, 22]

Az összekapcsolás részeként harmonizálni kellett a két erőforrás között az elemek megszorításait leíró ontológiákat, melyek között egy áthidaló fogalmakat tartalmazó ontológiával teremtettem meg az átjárhatóságot.

7. Tézis. *Kialakítottam egy ontológiát, amely összekapcsolja a magyar nyelvű MetaMorpho igéinek leírását az angol VerbIndex szintaktikai és szemantikus kategóriáival.*

A tézist alátámasztó közlemények: [11, 12, 4]

Össze lehetett kapcsolni a magyar és az angol nyelvű WordNeteket is, ezeket a kapcsolatokat is latba vetettem, hogy javítsam a minőséget, de ezen kapcsolatok minősége nem bizonyult megfelelőnek a feladat szempontjából.

8. Tézis. *Méréssel kimutattam, hogy a magyar és angol nyelvű WordNetek bevonásával nem lehet a fenti ontológia minőségét tovább javítani.*

A tézist alátámasztó közlemények: [11, 12, 4]

DOI:10.15774/PPKE.ITK.2018.007

A fejezetben ismertetett munka alkalmazható például a magyar szemantikai elemzés pontosítására, valamint jó minőségű szemantikai információkat tartalmazó igei adatbázisok előállítására, melyet az elméleti nyelvészet tud hasznosítani. Távlati célként az ontológiák alkalmazási területei között szerepel több, az angol nyelvű erőforrásokból elérhető nyelvfüggetlen információ megbízható, automatikus átemelése magyar nyelvre, azonban várhatóan a WordNet és a hozzá hasonló kézzel készített erőforrások a neurális hálók előretörésével háttérbe fognak szorulni, illetve új erőforrások fognak a helyükbe lépni, így ennek hasznossága kétséges.

Jelenleg egy nagy pontosságú és az eddigieknél nagyobb fedésű igei erőforráson (Manócska, lásd a 4.2.2. fejezet) dolgozom, mely szintaktikai szempontból empirikusan alátámasztható, statisztikai információt is tartalmaz, és ennek fokozatos bővítését tervezem szemantikai információkkal a leírtak alapján. Jelenleg nem látok esélyt az általam készített ontológia széleskörű felhasználására, annak a szabályalapú rendszerek miatti erős függése miatt.

DOI:10.15774/PPKE.ITK.2018.007

5. fejezet

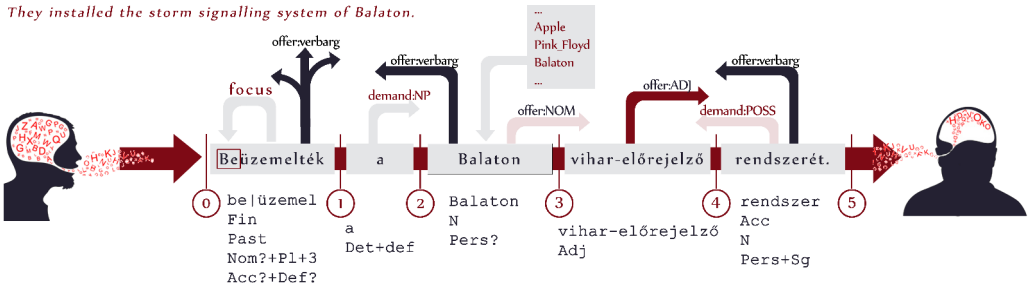
A pszicholingvisztikailag motivált elemző architektúrája

> „Így kell lennie: hogyan tévedhetne a halhatatlan, kollektív agy? Milyen külső mértékkel lehetne ellenőrizni a Párt ítéleteit? A józan ész statisztika dolga. Csupán arról van szó, hogy meg kell tanulnia úgy gondolkozni, ahogy ők gondolkoznak. Csak...!"
>
> (George Orwell: 1984)

5.1. Bevezetés

Az Anagramma elemzőmodell az 1.5.1. fejezetben már bemutatott elméleti felépítéséből adódóan a bemenetként kapott, akár több mondatot tartalmazó megnyilatkozáson balról jobbra, szavanként halad végig, klasszikus értelemben vett mondatrabontás és tokenizálás nélkül. A tokenekkel egyben kezeli központozást, ami ilyenkor új jellemzőket fűz a tokenhez, de nincs hatása a szótő kiszámítására és a lexikális szabályok illesztésére. A bemenet elején és amikor az elemzés eléri a bemenet végét, szintén definiál egy-egy határt, hogy a keresőeljárások ne tudjanak túlfutni az inputon. Így a rákötött beszédfelismerővel együtt teljességgel képes szimulálni a hallott szöveg, vagy a hírcsatornákon a képernyő alján végigfutó hírszalag emberi elemzőhöz hasonló feldolgozását (lásd az 5.1. ábra). Ebben a fejezetben a gyakorlati működést és a kezelt nyelvi jelenségeket ismertetem.

DOI:10.15774/PPKE.ITK.2018.007



5.1. ábra. Az ANAGRAMMA elemző vázlatos működése (Prószéky és Indig 2015b).

5.2. Alapfogalmak

Az elemzőben a tokenek és az általuk biztosított jellemzők – melyek a kínálatokat alkotják – egy attribútum-érték mátrixszal vannak reprezentálva. Kötelező elemük a szóalak, a szótő és az elemzés során kapott jegyek, melyek két csoportra oszlanak aszerint, hogy egy- vagy többértékűek lehetnek. Míg az előbbire példa a szám és személy, addig az utóbbira jó példa a *főnevet módosító (NPMod)* jegy (Vadász és Indig 2018). Az egyértékű jellemzők halmazként vannak reprezentálva és így halmazműveletekkel vizsgálhatók, míg a többértékűek az unifikáció megszokott módján (lásd *unifikáció-szerű eljárás*).

Ablaknak nevezzük azt az aktuálisan elemzett szótól jobbra eső, néhány szavas egységet, amely az utoljára elhangzott szóval ér véget. Az elemzés ilyenfajta „késleltetése" a lokális többértelműségek kiszűrésére szolgál. Az ablakról részletesen írok az 5.6. fejezetben.

Tározónak nevezzük azt az átmeneti munkamemóriát, amelyben az ANAGRAMMA az emberi elemzőhöz hasonlóan a már elhangzott, megelemzett részszerkezeteket tárolja. Ezen felül a tározóba kerülnek azok a keresőeljárások, amik az ablaktól jobbra keresnek.

DOI:10.15774/PPKE.ITK.2018.007

Keresőeljárás formájában definiáltuk az elméleti szinten megjelenő keresleteket, melyeket speciális attribútum-érték mátrixokban tárolunk. A morfológiai elemzésből képzett jellemzők indíthatnak keresőeljárásokat vagy azokat megszorító utasításokat[1], melyeket speciális keresőeljárásokkal definiáltunk. Ezek összekapcsolódhatnak és indíthatnak továbbiakat is. Működéséről részletesen az 5.4 fejezetben írok.

Unifikáció-szerű eljárással vizsgáljuk meg az elemzőben, hogy két elem (kereslet és kínálat, vagy kereslet és kereslet) kompatibilis-e. A megszokott unifikációval vizsgálható elemeken túl a keresési feltételek tartalmazhatnak szigorú egyezést elváró elemeket, valamint néha halmaz-értékű elemeket szükséges összevetni skalár-értékű elemmel, ebben az esetben a tartalmazást vizsgáljuk. Az unifikáció fogalmának ilyen irányú kiterjesztése miatt használom az „unifikáció-szerű" kifejezést.

Órajelnek az elemzőben azt tekintjük, amikor egy új elem előhívja a lehetséges keresleteit és kínálatait. Ez általában tokenenként történik, de fontos megjegyeznünk, az emberi elemzőhöz hasonlóan, a rendszer a bemenet soron következő tokeneit a meglévő tudása alapján megpróbálja illeszteni egy már ismert sorozatra, mely feldolgozható egy órajel alatt (lásd az 5.5. fejezet). Az órajelet felosztjuk több szakaszra, amelyek a feldolgozás különböző fokozatait, gyakorlatilag a keresés irányát jelentik. Minden szakasz után az újonnan létrejött elemek kettesével, párhuzamosan unifikálódnak egymással, míg a folyamat véget nem ér.

Határnak nevezzük azt az elemet, amelyet az egyes tokenek tudnak kirakni a jellemzőik által azért, hogy jelezzék bizonyos szerkezetek kezdetét vagy végét a keresőknek. Például egy minimális NP a determinánstól az esetragig tart, ezért célszerűnek látszik, hogy az első eleme magától balra megvizsgálja, hogy ő az NP első eleme-e, és pozitív válasz esetén kirakja a határt. Az NP fejének igényeit kielégítő keresőeljárás, amely a determinánst és a módosítókat keresi, így csak a

[1]Keresőeljárást megszorító esemény lehet az ige vonzatkeretének keresésekor például az a tény, hogy az ige participium formájában jelenik meg a tagmondatban, ekkor argumentumai csak tőle balra helyezkedhetnek el.

DOI:10.15774/PPKE.ITK.2018.007

határig kell, hogy elmenjen. A determináns megléte esetén a kérdés triviális, de annak hiányában az igenevek argumentumai miatt az állapottér igazán bonyolulttá válik. Ezzel az eljárással tehát a nagyobb ugrások minimalizálhatók, mely jellemző az emberi elemzőre is.

5.3. A hierarchikus jegyrendszer

Az elemző működésének alapját egy speciális hierarchikus jegyrendszer (Indig és Vadász 2016a) és egy kereslet-kínálat elven működő elemzési mód tesz ki. Az egyes szavak elemzéseiből olyan atomi jellemzőket állítunk elő, amelyek párhuzamos feldolgozása lehetővé teszi, hogy az egyes szavak a közvetlen szerepüket úgy tudják betölteni a mondatban, hogy akár több, egymásnak látszólag ellentmondó funkcióval is rendelkeznek[1]. A főnévi csoportot módosító szavak osztálya több különböző szófajból áll, melyeket így sajátosságaik figyelembe vételével, mégis egységesen tudunk kezelni az **NPMod** jegy által, az 5.1. táblázatban látható módon.

melléknév:	**CAS/Nom: tő+NPMod+Adj**	**+Sg/Pl(+PersSg/Pl1-3)**
számnév:	**CAS/Nom: tő+NPMod+Num**	**+Sg/Pl(+PersSg/Pl1-3)**
folyamatos melléknévi igenév:	**CAS/Nom: tő+NPMod+PartPres**	**+Sg/Pl(+PersSg/Pl1-3)**
befejezett melléknévi igenév:	**CAS/Nom: tő+NPMod+PartPast**	**+Sg/Pl(+PersSg/Pl1-3)**
beálló melléknévi igenév:	**CAS/Nom: tő+NPMod+PartFut**	**+Sg/Pl(+PersSg/Pl1-3)**

5.1. táblázat. A főnévi fejet módosító elemek lehetséges címkéi.

Az egyes módosítók kínálatként vannak jelen a rendszerben, ha egy fej magától balra keresné őket, viszont lehetnek maguk is az NP fejei az 1.4.1. fejezetben bemutatott módon, ami miatt saját keresőt kell, hogy indítsanak ennek tisztázására. Ettől teljesen független módon tud működni az igenevek vonzatainak, valamint az opcionális birtokos ragozásnak a kezelése, melyeket a továbbiakban részletesen bemutatok.

[1]Például az igenevek egyfelől a főnév módosítójaként is tudnak viselkedni, másfelől viszont saját vonzatkeretük van, és – bár megszorítottan – igeként is viselkednek.

DOI:10.15774/PPKE.ITK.2018.007

5.4. A keresőeljárások elemei

A kereső neve és indító tokenjének címe külön elemet alkot, mely nagyban segíti az elemző működésének nyomon követését, a hibák keresését. Továbbá, míg a név a behúzandó függőségi élek címkéjét adja, az indító elem a keresés végén új jellemzőkkel is gazdagodhat.

A keresés feltétele egy attribútum-érték mátrix, mely az illeszkedő token tulajdonságainak megszorításait tartalmazza. A megszorítás történhet a token *fő szófaji címkéje (főkategória)*, az egy- és többértékű jellemzői, valamint a szótő alapján is. A megszorítások lehetnek halmazértékűek is, mely esetben egy másik halmazzal szemben a metszet ürességét, skalárral szemben pedig a halmaztartalmazást vizsgálja a program. Amennyiben a főkategória értéke tetszőleges, akkor az illeszkedésnek pontosnak kell lennie a megadott feltételekre vonatkozóan, azaz a klasszikus unifikációtól eltérően nem megengedhető egyik operandusban sem olyan elem, amely nem szerepel a másikban.

Az irány azt szabja meg, hogy az adott igény az őt indító szótól melyik irányba keressen (balra, jobbra, az ablakban). Ezt azért fontos megkülönböztetni, mert a többértelműségek kiszűrésének elsődleges eszköze, hogy azok az elemek, melyek több irányba is kereshetnek, a megfelelő sorrendben „járják be" ezeket az irányokat.

Egyedi kínálatot keres egy kereső akkor, ha csak egy darab egyező elemet keres. Megfogalmazódik olyan igény – például az NP módosítók keresésénél –, ahol az összes ugyanolyan illeszkedő elemet meg kell találni az elemzés során. Ezért szükséges számon tartani a kereső ezen tulajdonságát.

A határ és a maximális távolság azt mondja meg a keresőnek, hogy meddig tud elmenni az adott irányban. A határral korlátozható például, hogy az ige az argumentumait a mondathatáron túl is keresse-e, a maximális távolsággal pedig beállítható, hogy maximálisan – amennyiben nem volt határ addig – a paraméterként változtatható darab token távolságra keressen csak. Az utóbbi lehetőség igen finom hangolást tesz lehetővé.

DOI:10.15774/PPKE.ITK.2018.007

A találati függvény akkor fut le, amikor az adott kereső talál egy egyező elemet, vagy az adott irányba történő keresés határba ütközik. Célja annak meghatározása, hogy mit csináljon a keresőeljárás, ha megtalálta illetve nem találta meg a keresett elemet: (a) indítson egy másik típusú keresést, (b) az azonos típusút folytassa vagy (c) fejeződjön be. Ha az azonos típusú keresést folytatja, akkor megváltoztathatja az irányt, vagy találat esetén elmehet a határig. A fentiektől függetlenül a keresést indító elem beállíthat jellemzőket magán, vagy húzhat függőségi éleket a talált elemre vagy akár saját magára is.

Ballasztnak neveztem el azt az elemet, amiben tárolhatóak azok az információk, amikre a kereső a működése folyamán vagy utána szükség lehet. Például az elvált igekötő megtalálása után a vonzatkeret lehívásakor azt az információt szükséges figyelembe venni, hogy az adott ige finit-e vagy sem, mert ez az argumentumok keresőinek irányát befolyásolja. A ballaszt további speciális felhasználási módjait az egyes nyelvi jelenségek bemutatásánál részletesen tárgyalom.

5.5. Az elemző egy órajele

Az elemzőprogram az aktuális tokent megpróbálja illeszteni a lexikonjában található „többszavas kifejezésekre", és egyezés esetén addig vizsgálja sorban a rákövetkező tokeneket, ameddig negatív eredmény nem születik. A továbbiakban pedig a maximális egyezést, amely egy teljes lexikális elemnek felel meg – figyelembe véve azt, hogy a lexikonbeli elemek utolsó tokene a megengedett módon (többnyire ragozás miatt) eltérhet –, annak hiányában pedig az eredeti tokent tekinti a bemenetről jövő következő elemnek, annak minden, a lexikon által definiált tulajdonságával együtt. Ennek a jelenségnek prototipikus esetei a több szóból álló, általában idegen nyelvi frázisok„ melyek részei külön nem feltétlenül értelmesek (pl. *Pink Floyd*) és személyről elnevezett intézménynevek (pl. *Petőfi Sándor Utcai Általános Iskola*), melyekben közös, hogy a lexikonhoz képest csak az utolsó token térhet el a ragozás miatt (pl. *Pink Floyddal*, *Petőfi Sándor Utcai Általános Iskolába*). Továbbá ilyen szerkezetek még a *gestaltok*, a lexikalizálódott, több tokenes szintaktikai szerkezetek, melyeket elemzés nélkül, *egészleges feldolgozással* kezel az emberi elemző (Pléh 1999).

DOI:10.15774/PPKE.ITK.2018.007

Az elemző ezután az aktuális szó jellemzőiből kinyeri a keresőeljárásokat, melyek unifikálódnak egymással és megindul a keresés az ablakban, balról jobbra, szavanként. Az ablak végére érve, az aktuális szótól balra folytatott keresést a tározóban lévő elemek jobbról balra (visszafelé) történő vizsgálatával folytatja. Ezek után az aktuális tokent berakja a tározóba, hogy az aktív (jobbra) keresők megvizsgálják illeszkedés szempontjából. Végül a megmaradt aktív keresőket is berakja a tározóba, ahol azok unifikálódnak a többi keresővel. Az egyes részfolyamatok végén a soron következő szó vizsgálatával kezdődik az új órajel.

5.6. Az ablak

Az emberi feldolgozás modellezésére az irodalomból jól ismert *Sausage Machine* kétfázisú mondatfeldolgozási modellt (Frazier és J. D. Fodor 1978) vettük alapul. A modellben az első fázis, az úgynevezett *Preliminary Phrase Packager (PPP)* az aktuálisan feldolgozott szó környezetében szereplő lokális többértelműségek feloldását és a szerkezetek „összecsomagolását" végzi. A nagyobb egységeket ezek után a második fázisban a *Sentence Structure Supervisor (SSS)* kapcsolja össze úgy, hogy közben ügyel az egymástól távolabbi többértelműségek helyes feloldására.

Az AnaGramma elemzőben használt ablakot a PPP fázis mintájára hoztuk létre. Úgy gondoljuk, hogy némi késleltetés illetve előretekintés olvasás közben feltétlenül szükséges az emberi elemző számára[1]. Ennek vizsgálatára létrehoztunk egy olyan weboldalt, amely a hírcsatornákon a képernyő alján végigfutó hírszalagot szimulálja[2], ahol az olvasás közbeni introspekcióinkat alapul véve megismételhető az intuitív megfigyelésünk: a következő néhány szó ismerete nélkül döntést hozni nehezünkre esik.

A PPP fázis angol nyelven egy „körülbelül hat szó méretű ablakon" működik. Ennél pontosabban a szerzők nem határolták be az ablak méretét, és nincs tudomásunk egzakt gépi megvalósításról sem. A magyar nyelv agglutináló jellege miatt az AnaGramma elemzőben három token méretű, flexibilis ablakot válasz-

[1] Az előretekintés és késleltetés jelenlétére utal, hogy a 2. fejezetben bemutatott módszerek jellemzői is felhasználják a jobb kontextust, bár nem valós időben.

[2] `http://users.itk.ppke.hu/~yanzigy/olvaso/`

DOI:10.15774/PPKE.ITK.2018.007

tottunk[1], mely szükség szerint „átugorhatja" az érdektelen elemeket, például a rövid, önálló jelentéssel nem bíró funkciószavakat.

Az ablak az elemzés szempontjából tehát azt jelenti, hogy a balról jobbra feldolgozás közben a baloldali kontextuson kívül az aktuális szó mellett annak két szó terjedelmű, jobboldali környezete is elérhető. A keresőeljárások az ablak bal oldalát jelentő, aktuális szótól indulva maximálisan az ablak jobb oldaláig tudnak elmenni. Amennyiben ez a környezet illeszkedő elemeket eredményez, azokat felhasználja az elemző az állapottér csökkentéséhez (pl. jelentős szerepük van a szófaji egyértelműsítésben is). Az ablakban található elemek csak a morfológiai információjukkal vannak jelen, az ő kereső és jellemzőállító eljárásaik csak akkor fognak elindulni, ha az adott elem az ablak bal oldalára kerül. Ebben a fázisban tehát nincs lehetőség mély elemzésre.

A továbbiakban két nyelvi jelenséget mutatok be, melyek az ablakkal szoros összefüggésben vannak. Korpuszokon végzett mérésekkel igazoltuk[2], hogy a vizsgált szerkezetek tekintetében az ablakból elérhető jobb környezet hordozza az elemzéshez elengedhetetlen információkat. A mérések ismertetése után bemutatom az azokból levont következtetéseket és a nyelvi jelenségeknek a levont tanulságokra épülő, az elemzőben megvalósított kezelését is.

5.7. Korpuszmérések

Az alábbi fejezetekben két különböző típusú, korpuszon végzett mérést fogok ismertetni az általam az *ablakkal* kezelni kívánt nyelvi jelenségekre. Az *előzetes mérések* célja az, hogy ismeretet lehessen szerezni a korpuszban valóban előforduló nyelvi fordulatokról – amelyek feldolgozása az elemzőrendszer fő prioritása –, a *kiértékelés* célja pedig az, hogy a megszerzett ismeretekből felállított modell helyességét számszerűen meg lehessen állapítani[3].

[1] Az ablak méretére és flexibilitására vonatkozó állítások az AnaGramma elemzőn túli, az emberi elemzőre való kiterjeszthetőségéhez szemmozgáskövetővel végzett kísérletek szükségesek, ám magyarra – eddig – ilyen kísérletről nem tudunk.

[2] A bemutatott korpuszméréseket Kalivoda Ágnes végezte. A nyelvi jelenségek kezelésének elméletét Vadász Noémivel közösen hoztam létre. A több nyelvi jelenségre kiterjesztett eljárás megtervezését és vizsgálatát Ligeti-Nagy Noémivel, Dömötör Andreával és Vadász Noémivel közösen végeztük.

[3] A méréseket és a nyelvi jelenségek vizsgálatát – azok összetettsége miatt –, több szerzőtársammal együtt végeztem. A dolgozatban a célom az ablak hatékonyságának valódi nyelvi

DOI:10.15774/PPKE.ITK.2018.007

5.7.1. A jelöletlen birtokos és birtokának relatív távolsága

A pontos feladat meghatározása *a testes esetragot magukon nem viselő névszók (tulajdonnevek, köznevek, melléknevek, számnevek, melléknévi igenevek) mondatbeli szerepének azonosítása*[1], ámbátor az előzetes mérések közül csak azokat az eseteket ismertetem, amelyekben a jelöletlen esetű szó a „valódi" nominatívusz, illetve a datívusz és a genitívusz esetek között alternál. A többi idetartozó eset korpuszbeli mintáinak vizsgálata hasonlóan történt(Ligeti-Nagy, Vadász, Dömötör és Indig 2018).

A magyarban kétféle birtokos szerkezet különböztethető meg (Bánréti et al. 1992): (1) a *jelöletlen birtokos* és (2) a *-nAk ragos birtokos*. Az utóbbi felismerése a datívusz miatt egyszerűbb – mivel nem esik egybe annyi esettel, mint a nominatívusz –, viszont a birtokától függetlenül bárhol helyet foglalhat a mondatban, ennek következtében ablakkal nem kezelhető. A jelöletlen birtokos esetében viszont a birtok jelölt, míg a birtokos nominatívuszi alakban áll, bár valójában genitívuszként viselkedik. Továbbá a birtoknak számban és személyben egyeztetve kell lennie a birtokosával, valamint a birtok csak a birtokos után következhet úgy, hogy nem kerülhet be közéjük a birtok névelője.

Ebben az esetben az elemző szempontjából tehát minden nominatívuszban álló elem potenciálisan birtokos, amíg az ellenkezője be nem bizonyosodik róla. Viszont ez csak akkor derül ki az elemző számára, ha megtaláljuk tőle jobbra a birtokoshoz tartozó, jelölt birtokot vagy egy jellemzőt, amely a birtokosságát cáfolja. A korpuszméréssel azt vizsgáltuk, hogy milyen messzire távolodik el a birtokos, mert ez segíti a nominatívuszi elem szerepének egyértelműsítését a megfelelő méretű ablakban.

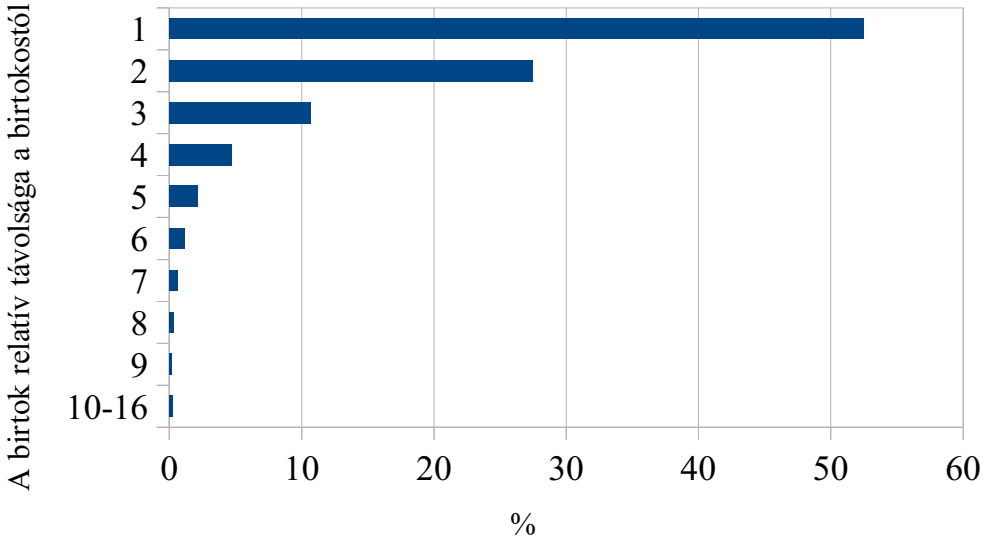
A Pázmány Korpuszban több mint 7 700 000 olyan szerkezetet találunk, amely megfelel a kritériumainknak (Indig, Vadász és Kalivoda 2016; Vadász és Indig 2018). Az 5.2. ábrán látható, hogy a birtok az esetek 52,46%-ában közvetlenül követi a jelöletlen birtokosát. 27,45%-ban egy token ékelődik közéjük (általában

jelenségek kezelésével történő alátámasztása. A nyelvi jelenségek és az elemzőbeli kezelésük részletes ismertetése túlmutatnak a dolgozat keretein.

[1]A mondatbeli szerepen azt értem, hogy az adott névszó a mondatban vonzat, szabad határozó vagy egyéb funkciója van egy főnévi csoporton belül.

DOI:10.15774/PPKE.ITK.2018.007

melléknév vagy számnév), 10,66%-ban két token (általában egy határozószóval módosított melléknév vagy számnév) kerül a birtok és birtokosa közé.



5.2. ábra. A birtok relatív távolsága a jelöletlen birtokosától. A függőleges tengelyen a tokenben mért relatív távolság, a vízszintes tengelyen a százalékos eloszlás látható.

Kevesebb, mint 10%-ban fordul elő, hogy a birtok ennél messzebbre kerül a jelöletlen birtokosától. Az utóbbi esetben általában egy frázis, egy melléknévi igenév és vonzatai vagy felsorolás által felduzzasztott NP kerül be[1] a birtokos és a birtoka közé, mint az a (19) példában is látható, ahol **vastag** betűvel emeltem ki a jelöletlen birtokos szerkezet birtokát és birtokosát.

(19) ***Krisztina*** *különleges , Swarovski kristályokból és minőségi japán gyöngyökből készült , egyedi tervezésű romantikus , nőies* ***nyaklánca***

Következtetésként elmondható tehát, hogy a jelöletlen birtokos birtoka az esetek kb. 80%-ában a 3 szélességű ablakon belülre esik, és az ablak ezért segíteni tudja a birtokos tisztázását. Az erre vonatkozó eljárást részleteiben az 5.8 fejezetben ismertetem. A maradék 20%-ban is többnyire csak egy nagyon kibővített

[1]Ez a jelenség mutatja, hogy az 1.4.3. fejezetben bemutatott MaxNP konstrukciók a való életben nem csak egymástól függetlenül, hanem keveredve is előfordulnak.

DOI:10.15774/PPKE.ITK.2018.007

szerkezet kerül a birtokos és a birtoka közé, mely információ az ilyen szerkezetek más módon történő kezelését könnyíti meg. Feltételezzük, hogy ilyen esetekben az emberi elemző egy másik, még fel nem térképezett stratégiát használ, mely vizsgálatára további kutatás szükséges.

5.7.2. Az elváló igekötő távolsága

Minden igének és igéből képzett szónak lehet vonzatkerete és igekötője. A többféle lehetséges vonzatkeret mielőbbi egyértelműsítése csökkenti az állapotteret és az elemző működését gyorsítja. Ebben fontos szerepet játszik az igekötő és az infinitívuszi vonzat[1], ugyanis jelenlétük ismerete vagy cáfolata különféle megszorításokra ad lehetőséget az ige vonzatkeretében. Az elvált igekötő jellemzően nagyon közel áll az igéjéhez, valamint az infinitívusz sokszor szerepel közvetlenül az ige után. Az ilyen módon „későn érkező" elemek megtalálása az ablak segítségével viszont nagyban segíti a balról jobbra elemzést.

A mérések három különböző korpuszon történtek. Az MNSZ 2.0.3 és a Pázmány Korpusz nyelvmodellként szolgált, mivel egyaránt vannak bennük szerkesztett és szerkesztetlen szövegek. Az InfoRádió Korpusz pedig ideális bemenetnek tekinthető, mivel csak jól szerkesztett rövidhíreket tartalmaz. Mindhárom korpusz sok hibás annotációt tartalmaz, ezért a rossz találatok automatikus kiszűrésére egy több mint 27 ezer igekötős igelemmát tartalmazó (manuálisan ellenőrzött) listát (Kalivoda 2016) használtunk fel. A kapott találatok közül csak azokat őriztük meg, amelyek esetén az igekötő–ige pár a listában szereplő kombinációk egyike volt. Ezzel kiszűrődnek az olyan, listában nem szereplő, de amúgy helyes találatok is, mint például a neologizmusok, de a mérésben a pontosság sokkal fontosabb volt, mint a fedés. A lista továbbá arra is alkalmas volt, hogy segítségével eldönthető legyen, hogy az igekötő a finit vagy az infinit igéhez tartozik (esetleg tartozhat-e elvben mindkettőhöz). Minden esetben a két elem egymáshoz képesti pozíciója lett összehasonlítva, melyben a 0 pozíció az ige, és így a tőle közvetlenül jobbra álló elem a +1 pozícióban van.

[1] A dolgozatban az igekötő és az infinitívusz vonzatkeret-egyértelműsítő szerepe van a fókuszban, a további ígéretes jellemzők (speciális vonzatsorrend, lexikális kötöttség, stb.) felkutatása nyitott kérdés.

DOI:10.15774/PPKE.ITK.2018.007

5.7.2.1. Finit igék posztverbális igekötői

Először a finit ige és a tőle jobbra került igekötője távolságát vizsgáltuk meg (lásd az 5.2. táblázat). Bár az ige utáni fő összetevők sorrendje a mondatban alapvetően szabad (É. Kiss 2007), az MNSZ2-n végzett mérések (Indig és Vadász 2016b; Kalivoda 2016, 2017) azt mutatták ki, hogy a posztverbális igekötők az esetek 99%-ában +1 vagy +2 pozícióban állnak. Ezzel szemben az InfoRádió Korpuszban ez az érték 100%, vagyis nincs benne példa olyan esetre, ahol egynél több szó áll a finit ige és annak posztverbális igekötője között. Ez igazolja azt a feltételezésünket, hogy a hivatalos, szerkesztett szövegek formája kötöttebb.

FIN	+1	+2	+3	+4	+5	+6	+7
MNSZ2	7 527 308	163 993	5 126	1 193	267	101	27
InfoRádió	23 552	220	-	-	-	-	-
MNSZ2 (%)	97,778%	2,130%	0,066%	0,015%	0,003%	0,001%	3,5e-4%
InfoRádió (%)	99,999%	0,001%	-	-	-	-	-

5.2. táblázat. A finit ige és posztverbális igekötőjének távolsága – szerkesztett szövegekben 99,999%-ban közvetlenül az ige után, szerkesztetlen szövegekben 99,9%-ban maximum két token távolságra helyezkedik el az igekötő.

Az eredmények tehát azt mutatják, hogy az ige utáni igekötő nagyon ritkán kerül 2 tokennél távolabbra az igéjétől, és az általunk javasolt elemzési ablakba éppen belefér. Az eltávolodott igekötőket tartalmazó, ritka példákról elmondható, hogy saját szavakkal történő felidézésükkor az igekötő többnyire az eredeti mondatbeli helyénél közelebb kerül az igéhez. Az ilyen mondatokra példa az alábbi két mondat, amely az MNSZ2-ből származik:

(20) Azért **mentem** egy kicsit a pop zene fele **el**, mert szeretem a nívós, könnyed jó popzenét.

(21) 27 gyereket **vitt** egy feltehetően részeg buszsofőr Szentesen még csütörtökön egy sportrendezvény után **vissza** az iskolába.

DOI:10.15774/PPKE.ITK.2018.007

Az igekötők típusainak tekintetében megfigyelhető, hogy a legtávolabb az *egynél több szótagból álló igekötők* kerülhetnek, mert határozószóként is funkcionálnak, viszont pont ezért az ilyen igekötők nem befolyásolják számottevően az ige vonzatkeretét. Ebbe a csoportba tartozik például a *haza* és a *vissza*. Az ige közelségében leginkább rövid, prototipikus igekötők fedezhetők fel, amelyek eloszlásukat tekintve hasonlóak[1] (lásd az 5.3. táblázat[2]).

	-2	0	+1	+2	+3
meg, ki, be, le, fel, föl, el, át, rá	0,49%	58,5%	40%	1%	0,01%

5.3. táblázat. Néhány gyakori igekötő távolsága a finit igétől – az igekötők 98,5%-a az igén vagy közvetlenül utána áll, csak 1,01% távolodik el jobban jobb oldalra (MNSZ2).

5.7.2.2. Az infinitívusz és a posztverbális igekötője

Az infinitívusszal kapcsolatos mérések a Pázmány Korpuszon történtek, mely webalapú korpusz révén még az MNSZ2-nél is több szerkesztetlen szöveget tartalmaz (a kommentkorpusz mérete 2 millió token). Az eredmények mégis azt mutatják, hogy az igekötő az esetek 86%-ában közvetlenül az infinit igealak után áll (lásd az 5.4. táblázatot).

Az eredményekből láthatjuk, hogy a kiugróan gyakori +1 pozícióban még sok prototipikus igekötőt találunk, pl. *iparkodott ellentétet* ***mutatni ki***, *javasolt a lapokat lazán* ***helyezni el***. A +2 pozícióról elmondható, hogy az infinitívusz és annak igekötője között csak finit ige állhat, és – bár van példa prototipikus igekötőre, pl. *épp* ***foglalni*** *akartam* ***le*** *a buszt* – nagyobb arányban jelennek meg a nem

[1] A preverbális igekötők pozíció szerinti eloszlását lásd (Kalivoda 2016).

[2] Az 5.3. táblázatban azért nem szerepel a -1-es pozíció, mert a magyar helyesírás szerint egy egytagú igekötő nem előzheti meg közvetlenül az igét, amelyhez tartozik. Ebben az esetben egybeírandó az igével (0. pozíció). Az igét közvetlenül megelőző pozícióban szereplő igekötő általában egy másik igei elem (pl. az ige utáni infinitívusz) igekötője, és csak elírás eredményeképpen lehet az igéé.

DOI:10.15774/PPKE.ITK.2018.007

INF [...] IK	db.	%
össz.	717	
+1	619	86,3
+2	52	7,3
+3	35	4,9
>+3	11	1,5

5.4. táblázat. Az infinitívusz és a tőle jobbra elhelyezkedő igekötőjének távolsága – 93,6%-ban maximum két token van köztük.

FIN [...] INF	db.	%
össz.	727 562	
+1	652 778	89,7
+2	47 669	6,6
>+2	27 115	3,7

5.5. táblázat. A finit ige és a tőle jobbra elhelyezkedő infinitívuszi vonzatának távolsága – 96,3%-ban maximum két token van köztük.

prototipikus, több szótagú igekötők (pl. *már **indulni** akartam **vissza***). A nagyon ritka +3 pozícióban csak ez utóbbiak állnak, pl. *de már **jönni** kellett sajnos **haza***. A +4 és +5 pozícióra mindössze 15 példát találtunk, ami statisztikailag irreleváns mennyiség. Az itt álló igekötők nem befolyásolják az ige vonzatkeretét (csak az ige által kifejezett mozgás irányát módosítják), pl. ***vinni** kell a kamerát **el**, **menekülni** akartak a városon keresztül **vissza***. Megmértük továbbá az infinitívusz igétől való távolságát is. A 5.5. táblázatban látható, hogy az esetek 89%-ában az infinitívusz közvetlenül a finit ige után áll, 6,5%-ban egy szót enged maga elé. Tehát az infinitívusz többnyire benne van a főige mellett az ablakban, és felhasználható a vonzatkeret egyértelműsítésére.

A mérések eredményéből tehát látható, hogy a balról jobbra elemzés során a finit igén állva az ablakában a legtöbb esetben benne van a posztverbális igekötő és az infinitívuszi vonzat, mely ezáltal segíteni tudja a vonzatkeretének egyértelműsítését. Hasonlóan az infinitívusznál, melynek igekötőjére és infinitívuszi vonzatára is áll az előbbi megállapítás. A fenti mérések alapján létrehozott *VFrame* keresőeljárást, amely az igekötők igei elemekhez kapcsolásával segít előhívni a mondatban előforduló finit és infinit igék megfelelő vonzatkeretét, az 5.9.2. fejezetben ismertetem.

DOI:10.15774/PPKE.ITK.2018.007

5.8. Az NP-k kezelése az AnaGramma elemzőben

A mondatelemzés első fázisában a névszói frázisok összeállítása történik. Ha az NP-t egy determináns nyitja bal oldalról, az az elemzés során kínálatként a tározóba bekerülve várja a fej Det-keresőjét. A magukon testes esetragot nem viselő névszók megfigyeléseink alapján az alábbi szerepekben állhatnak[1], melyek közül az elemzőnek választania kell: lehetnek a mondat alanyai, egy jelöletlen birtokos szerkezet birtokosai, egy névutós szerkezetben a névutó „vonzatai”[2], főnevet módosító elemek, illetve a mondat névszói állítmányai vagy annak részei. A különböző mondatbeli szerepek alapján megállapíthatjuk, hogy a testes esetragot nem viselő névszó végén egy testetlen esetrag van (ha a névszó alanyesetben vagy jelöletlen birtokos esetben van), vagy nincs semmi (ha főnévi fejet módosít vagy egy névutó vonzata). A névszói állítmány szerepében álló névszók esetében (elsősorban technikai megfontolásból) szintén egy testetlen esetragot feltételezünk[3]. A dolgozatban ezen esetek jelölésére rendre a következő szimbólumokat használom: α, 0, β.

5.8.1. A `Nom-or-What` eljárás motivációja

A `Nom-or-What` eljárás[4] (Ligeti-Nagy, Vadász, Dömötör és Indig 2018) az AnaGramma kereteiben működő `Nom-or-Gen` eljárás (Vadász és Indig 2018) továbbfejlesztett és kiegészített változata, melynek a célkitűzése az, hogy egy testes esetrag nélküli elem esetében annak szűk kontextusa alapján eldöntse, hogy egy jelöletlen birtokos szerkezet birtokosa-e. Az eljárást több szerzőtársammal együtt úgy fejlesztettük tovább, hogy az alanyesettel, a főnevet módosító elemekkel és a

[1] A bemutatott eseteken kívül a teljesség kedvéért meg kell említeni, hogy noha a jelenlegi algoritmusban nincsenek kezelve és további kutatást igényelnek, előfordulhat még a *vokatívusz („megszólítás-eset”)* és az úgynevezett *többtagúnév-eset*, melyre példák találhatók Ligeti-Nagy, Vadász, Dömötör és Indig (2018) munkájában.

[2] A névutó előtti névszót a névutó vonzatának hívom egyszerűsítésképpen, de ez nem elméleti nyelvészeti megalapozottságú állásfoglalás. A névutó és az esetrag nélküli névszó korpuszbeli együttes előfordulására utal: nincs névutó névszó nélkül.

[3] A névszói állítmány esetében feltételezett esetrag a névszó „állítmány-esetét” jelöli.

[4] A korpuszokban a testes esetrag nélküli névszók esetcímkéje kivétel nélkül `NOM`, és az esetegyértelműsítő eljárás feladata tisztázni ezeknek az elemeknek az aktuális szerepét, ezért az eljárás a `Nom-or-What` nevet kapta.

DOI:10.15774/PPKE.ITK.2018.007

névutók vonzataival kapcsolatban is képes legyen döntést hozni, azonban egyéb tulajdonságaiban megegyezik elődjével.

Mindkét eljárás abban a megfigyelésben gyökerezik, hogy az NP módosítók több különböző szófajú elemből állhatnak, melyek az elemzőben egységesen az **NPMod** bennfoglaló kategóriába tartoznak (lásd az 5.3. fejezet), és kínálatként viselkednek az NP feje számára. Ugyanakkor az **NPMod** jegy saját keresőt is indít, hiszen el kell döntenie, hogy nem ő-e az NP feje, melyet az ablakban látható további módosító vagy főnév találásával, vagy hiányának feltárásával tud eldönteni, mely utóbbi sok esetre szétbontandó bonyolult döntést kíván.

A legegyszerűbb esetben, ha az NP feje egy főnév – és nem záródott le az NP korábban egy esetragos **NPMod** kategóriájú elemmel –, annak a jegyei az 5.6. táblázatban látható módon alakulnak.

főnév:	**CAS/Nom: tő+N(+PropN)**	**+Sg/Pl(+PersSg/Pl1-3)**

5.6. táblázat. NP-t lezáró főnévi fej jegyei

Az **N** jegy indítja el a determináns és a főnevet módosító tokenek (**NPMod**) keresését, mely eljárások így függetlenül tudnak működni az esetragok egyértelműsítésétől. A főneveknél továbbá különbséget kell tennünk a tulajdonnév és a köznév elemek között. Az előbbi egy opcionális **PropN** jegyet kap, mely módosítja a determinánskeresőt úgy, hogy ha nem található determináns, akkor is határozott legyen a főnév a tulajdonnév-jellege miatt, amit egy önmagára húzott függőségi éllel jelez a program. Ezen kívül a főnevek jegyei között is szerepel az egyes illetve többes számot kifejező jegy, valamint a birtokos ragozás jegye a birtokosnak megfelelő számmal és személlyel.

A névszók fenti egyszerűen kezelt tulajdonságaitól leválasztva kezelhető az esetrag. Mely ha testes (**CAS** jegy), legyen az egy **NPMod** jegyű elemen vagy egy főneven, minden esetben kirak egy határt az adott elem mögé, jelezve, hogy a frázis bezárult. Mivel az igei argumentumok esetragos elemet keresnek, az NP csak az esetrag megjelenése után fog tudni egy függőségi éllel az igéhez kapcsolódni. A nominatívuszi testetlen esetrag (**Nom** jegy) esetén fokozottan számítani kell arra, hogy valójában genitívusszal vagy névutós főnévvel van dolgunk. Ilyenkor az eset egyértelműsítéséig nem kapja meg a főnév a **CAS** jegyet.

DOI:10.15774/PPKE.ITK.2018.007

A hagyományosan NOM címkével jelölt, a fentiekkel egybeeső, állítmányi szerepű névszók esetében csak akkor tudna dönteni a rendszer, amikor a névszó szűk környezetében szerepel olyan információ, amely ezt lehetővé teszi. Ez azonban ritkán fordul elő, az állítmányi szerep felismerésében ugyanis jellemzően inkább a névszót megelőző mondatrész nyújt nagyobb segítséget. Az ablak alapján a névszói állítmányi eset csak akkor azonosítható egyértelműen, ha az ablakban létige található, sőt, még ezt is le kell szűkíteni 1-2. személyű létigére, egyébként nem lehetünk biztosak benne, hogy a létige valóban kopula. Míg a (22a) példában az ablak alapján egyértelműsíthető a névszói állítmányi eset, addig a (22b) és a (22c) példa esetragjainak elkülönítéséhez az előzmények ismeretére is szükség van. Még nehezebb a helyzet, ha egyáltalán nincs a mondatban testes kopula, ilyenkor két fő ismérvre támaszkodhatunk: 1) a mondatban nincs finit ige, 2) a mondatban már van nominatívusz. Mindkét jellemző csak a nagyobb kontextusból, jellemzően az állítmányi szerepű névszót megelőző mondatrészből állapítható meg.

(22) a. *Negyedik **gyerek** voltam a családban.*
gyerek/gyerek/FN.β

b. *Erdélyi Dániel maga is iskolás **gyerek** volt a film ábrázolta korszakban.*
gyerek/gyerek/FN.β

c. *Kevés zsidó **gyerek** volt a falumban.*
gyerek/gyerek/FN.NOM

Mindezek alapján azt feltételezzük, hogy az alanyesetet, a birtokos esetet, a főnévi módosító szerepet, a névutó vonzatát és bizonyos esetekben az állítmányszerepű névszókat is a kétfázisú mondatelemzés első fázisában tisztázzuk, amikor a mondat elemeit előkészítjük arra, hogy megkapják szerepüket a mondatban. Az egyértelműsítésben pedig az elemző közvetlen jobbra tekintő ablaka segít, mert az esetrag tisztázása itt is nagyban csökkenti az elemzés komplexitását[1]. Az elemzőben a névutós NP-k esetében a névutóval való összekapcsolódás után válik

[1]Várhatóan a baloldali kontextus felhasználásával tovább csökkenthető a komplexitás, de ez túlmutat az esetragok egyértelműsítésén.

DOI:10.15774/PPKE.ITK.2018.007

csak láthatóvá a frázis az igei argumentumkeresők számára[1]. A névszói állítmány teljes bizonyossággal történő megtalálása pedig a kétfázisú mondatelemzés második fázisában történik, hiszen felismeréséhez szélesebb kontextus szükséges. Jelen dolgozatban viszont csak az elemzés első fázisára szorítkozom, mivel a fentiekből látható, hogy a sok egybeeső eset helyes kezelése komplex megoldást kíván.

5.8.2. A `Nom-or-What` eljárás

Megfigyeléseink szerint külön szabályrendszert kell alkotni a főnevekre, a melléknevekre, a számnevekre és a melléknévi igenevekre, mivel jellemzően máshogy viselkednek, ami a testes esetrag nélküli tokeneiket illeti (Ligeti-Nagy, Vadász, Dömötör és Indig 2018). Mindegyik szófaji kategóriához két-két listát készítettünk: egyet azokról az esetekről, amikor az adott szófaji kategóriához tartozó, `NOM`-nak címkézett elem egy NP belsejében található, és azokról az esetekről, amikor az adott szófaji kategóriához tartozó, `NOM`-nak címkézett elem egy NP legutolsó eleme. Ezen listák alapján állapítottuk meg, melyek azok az elemek, amelyek előtt biztosan eldől egy `NOM` jelentése, és melyek azok, amelyek előtt nem. Megfigyeléseinket és az algoritmus működését döntési fák segítségével szemléltetem (F.2., F.3. és F.1. ábrák).

Mivel az algoritmus az ablakban szereplő tokenekre támaszkodva dönt, értelemszerűen lesznek olyan esetek, amikor nem egyértelműsítheti a testetlenséget. Ezekre az esetekre *alapértelmezett (default)* értéket kellett meghatározni: a főnevek (mind a köznevek, mind a tulajdonnevek) esetében az *alapértelmezett* érték az α, az alanyeset és a jelöletlen birtokos eset testetlen esetragja; a melléknevek, számnevek és a melléknévi igenevek esetében pedig a *default*_0, ami abban különbözik a 0-tól, hogy nem mond egyértelmű ítéletet, az adott `NPMod` kategóriájú elem még NOM vagy GEN címkét is kaphat később, a mondat egészének ismerete alapján. A 23a példában az *ember* token mondatbeli szerepe az ablakban látható elemek alapján nem állapítható meg biztosan, ezért α esetragot kap. A (23b) példában az *egyik* esete nem egyértelműsíthető az ablak alapján, ezért *default*_0 esetet kap.

[1]Kijelenthető, hogy a névutók mint esetet kifejező elemek (Indig és Vadász 2016a) az esetragokhoz hasonlóan kezelendők, hiszen a névutók is a névszói frázist lezáró elemek.

DOI:10.15774/PPKE.ITK.2018.007

(23) a. *Elindul reggel hazulról az **ember** ingujjasan vagy pulóverben.*
ember/ember/FN.α

b. *A Nagy_Szent_Bazil-rendnek két kolostora is működött, az **egyik** Veszprémben, a másik Dunapentelén.*
egyik/egyik/MN_NM.default_0

Az F.2. ábrán látható a főnevekre (köznevekre és tulajdonnevekre), illetve többesszámú melléknévnek, számnévnek, melléknévi igenévnek annotált elemekre vonatkozó szabályok összefoglalása döntési fában. A fa gyökere az aktuális elem szófaji címkéje. A fa első szintjének élein az ablakban látható első elemen található információk szerepelnek. Például az ablak első elemén látható NU címke következtében az algoritmus a 0 esetet ítéli meg az aktuálisan vizsgált token NOM esetragjának helyére. A fa második szintjén lévő élek az ablak második elemén látható információkat tartalmazzák: ezek csak akkor aktiválódnak, ha az első elem alapján az algoritmus nem tudott dönteni, és a *default* α esetet illesztette a NOM helyére. Ekkor az ablak második elemén látható bizonyos címkeelemek alapján még van lehetősége egyértelműsíteni az esetet. Az ábrán nem tüntettük fel, de a 0 egyértelműsítése után (a fa második ágának végrehajtása előtt) történik egy lépés, amely független az ablak első elemétől. Ha az aktuálisan vizsgált token semmilyen esetben sem lehet jelöletlen birtokos szerkezet birtokosa, akkor az algoritmus a NOM esetet ítéli meg a szónak, és megtörtént az egyértelműsítés. Ilyen tokenek a következők: *az, ez, mindaz, mindez, aki, ami*. Fontos kitétel, hogy az ablak második elemén látható birtokos személyjel (PS) csak abban az esetben vezet az aktuális elem NOM címkéjének GEN esetragra cseréléséhez, ha az aktuális elemen nincsen birtokos személyjel. Ezzel a *Magyarország **kormánya** mostani megbízásából* szerkezetek előfordulása zárható ki. A *Magyarország **kormánya** megbízásából*-típusúak az algoritmus számára is megítélhetőek, az ablak első elemén látható birtokos személyjel alapján. A példákban félkövérrel szedett tokenek az aktuálisan vizsgált elemek.

A legfelső élen, a NU társaságában található a NU_MN kategória. Bár ilyen címke az MNSZ2-ben nem található, mi fontosnak tartjuk a megnevezését: az *alatti, általi, mögötti* stb. szavak tartoznak ide. Ezek, a névutókhoz hasonlóan,

DOI:10.15774/PPKE.ITK.2018.007

azonnal egyértelműsítik az őket megelőző névszói elem végén a 0 esetet. Hozzájuk hasonló a *című, nevű* tokenek szerepe, ezért azokat is feltüntettük itt[1].

Az F.3. ábrán látható a(z egyesszámú) melléknevekre és melléknévi igenevekre vonatkozó szabályok összefoglalása döntési fában. Ennél a szabálycsoportnál is fontos kitétel, hogy az ablak második elemén látható birtokos személyjel (PS) csak abban az esetben vezet az aktuális elem NOM címkéjének GEN esetragra cseréléséhez, ha az aktuális elemen nincsen birtokos személyjel. Végül az F.1. ábrán látható az előzőekhez hasonlóan a számnevekre vonatkozó szabályok összefoglalása.

5.8.3. A Nom-or-What eljárás kiértékelése

Az algoritmus teljesítményét 1 000 darab tesztmondaton mértük ki. A tesztmondatokat az MNSZ2-ből vettük úgy, hogy a mondatban legyen legalább egy finit ige. Az így kapott 1 000 darab, véletlenszerűen kiválasztott mondaton további három változtatást eszközöltünk:

- Mivel a tulajdonneveknek jól körülhatárolható, fontos funkciója van az őket megelőző elem esetének egyértelműsítése során, ezért a tesztmondatainkon kézzel annotáltunk minden tulajdonnevet. A többelemű tulajdonneveket _ jellel összekapcsoltuk, és FN címkéjüket TULN címkére cseréltük.

- A kopulát vagy finit igét nem – legfeljebb létigét – tartalmazó tagmondatokat töröltük. Ha egy egész mondatot kellett a kopula miatt törölni, akkor ahelyett újat kértünk a korpuszból.

- Az esetlegesen bennragadt, „szemétnek” minősülő elemeket, úgy mint a sorszám a mondat elején vagy a végén, töröltük a mondattokenekből.

Az így megtisztított 1 000 darab mondat minden egyes, eredetileg NOM címkével rendelkező eleme három elemzést kapott: egyet az algoritmustól, egyet a kézi annotáció során az ablak elemeire támaszkodva, és egyet a kézi annotáció során mint végleges elemzés.

[1] Megjegyzendő, hogy a *című* és a *nevű* előtti eset inkább a többtagú nevek esetével azonos, de ez további kutatást igényel.

DOI:10.15774/PPKE.ITK.2018.007

A kétféle kézi annotáció segítségével egyszerre tudunk ítéletet mondani az algoritmus megvalósításáról, valamint az algoritmus elméleti keretéről, az ablakról. Az ablak alapján hozott manuális ítélet (lásd a (24) példában az α kimenet) lényege, hogy kiértékelhessük, hogy az algoritmus az ablakban elérhető információk alapján jól dönt-e. A teljes mondat figyelembevételével definiált kézi annotáció (lásd a (24) példában a NOM kimenet) pedig az adott tokenek végső szerepét mondja meg (a (24) példában a *patak* a mondat alanya). Ha a két kézi annotáció egyezik, azaz a teljes mondat alapján kapott kézi elemzés nem mond ellent annak, amit az ablak alapján a manuális annotáció során mondtunk, akkor csupán az ablakbeli információk alapján nagy bizonyossággal meg lehet határozni egy névszó mondatbeli szerepét, anélkül, hogy az aktuális token nagyobb környezetében körül kellene néznünk. Vagyis a kétfázisú mondatelemzés első fázisában ezek a szerkezetek jól egyértelműsíthetők.

(24) *A **patak** tőlük keletre húzódott.*
az ablak: *patak tőlük keletre*
az algoritmus ítélete: *patak FN.**nom***
kézi annotáció, az ablak alapján: *patak FN.*α
kézi annotáció, a teljes mondat alapján: *patak FN.**nom***

A kézi annotáció során, az ablak alapján való döntésnél a következő címkéket kaphatták a testes esetrag nélküli tokenek:

- NOM: nominatívusz
- GEN: birtokos
- 0: esetrag nélküli (névutó előtti névszó[1], vagy más névszó módosítója)
- α: nem eldönthető; a főnevek *alapértelmezett* értékét kapja (NOM-má vagy GEN-né egyértelműsödhet később)

[1]A névutó előtti névszó végén lévő testetlenség pontos jelentése nyelvészetileg kérdéses: egyfelől egy NOM jelenlétét feltételezhetnénk, az esetragos névszót vonzó névutók példáinak analógiájára, pl. *a kerítésen kívül – a kerítés mellett*; ugyanakkor a névutós és esetragos névszók analógiájának velejárója, hogy a névszó végén valóban nincsen semmi az esetrag előtt, így nem feltételezhetünk semmit a szótőn, pl. *az asztalon – az asztal alatt.* A kiértékelés során az utóbbi analógia mintájára a névutó előtti névszók végén nem feltételezünk NOM vagy más esetragot.

DOI:10.15774/PPKE.ITK.2018.007

- *default_0*: nem eldönthető; az `NPMod` elemek *default* értékét kapja
- VOK: vokatívusz esetű
- *postag_hiba*: valamelyik vizsgált elem hibás szófaji címkét kapott, ezért rossz az elemzés (például melléknévi igenévnek címkézett ige esetében)
- többtagú nevek esete (pl. *Tóth kisasszony, elnök úr*)

Míg a kézi annotáció során a teljes mondat alapján hozott ítéleteknél a következő címkét kaphatták az esetegyértelműsítésre váró névszók:

- NOM
- GEN
- 0
- VOK
- többelemű nevek esete
- α vagy *default_0*, abban az esetben, ha a teljes mondat kétértelmű

A tesztmondatokban összesen 125 olyan token volt, amelyeknél vagy az adott token annotációja volt hibás (például `NOM` esetragú melléknévi igenévnek volt címkézve egy ige), vagy az ablakban lévő egyik vagy másik szóalak (például az őt követő melléknévi igenév volt igének címkézve). Ezeket az eseteket nem javítottuk, nem számítottuk az értékelésnél. Szintén nem került bele a kiértékelésbe a 34 vokatívuszi esetű névszó, illetve a 45 darab *többtagúnév-esetű* token.

A kiértékelés szabályait és a kategóriákat az 5.7. táblázat tartalmazza. *Valós pozitív (True positive, TP)*, *álpozitív (false positive, FP)* és *álnegatív (false negative, FN)* kategóriákat állapítunk meg. Az egyes oszlopokat a következőképpen kell értelmezni: ha a „kiértékelendő eredmény" oszlopban látható értékre a „sztenderd" oszlop adott értéke illeszkedik, akkor ez egy TP, FP vagy FN találat, attól függően, melyik sorban található ez a párosítás. Ez a megfeleltetés igaz akkor is, amikor az algoritmus eredményét hasonlítjuk a *csak az ablakot figyelembe vevő kézi annotációhoz*, illetve akkor is, amikor a *csak az ablakot figyelembe vevő kézi*

DOI:10.15774/PPKE.ITK.2018.007

annotációt a *teljes mondatot figyelembe vevő kézi annotációhoz* hasonlítjuk. A TP eredmények a teljes egyezések. FP eredménynek tekintettük a túlspecifikálást: ha például az algoritmus egy elemről azt állítja, hogy nominatívusz, de a kézi annotáció szerint az ablak alapján még nem mondhatna ilyet, csak egy *default* α-t. Viszont ha alulspecifikál, tehát *default* értéket ad egy elemnek, pedig az ablakból eldönthető lenne pontosabban is, az FN.

kategória	a kiértékelendő eredmény	a sztenderd
TP	NOM	NOM
	GEN	GEN
	α	α
	*default*_0	*default*_0
FP	NOM	α
	GEN	α
	0	*default*_0
FN	α	NOM
	α	GEN
	*default*_0	0

5.7. táblázat. A kiértékelés szabályai.

5.8.3.1. Az ablak kiértékelése

Az 5.8. táblázatban látható eredmények azt mutatják, hogy a kételemű ablak alapján történő egyértelműsítés milyen pontosságot és fedést eredményez, ha a testes esetrag nélküli névszói elemek teljes mondat alapján történő esetegyértelműsítéséhez hasonlítjuk. Jól látszik, hogy a pontosság 97,73%-kal igen magas. Ennek oka, hogy az alapvető cél az volt, hogy az algoritmus precízen döntsön, és ne kelljen a mondatelemzés egy későbbi fázisában korrigálni a később tévesnek bizonyuló ítéleteket.

A fedés ugyanakkor csak 67,63%: ez legfőképpen a FN találatok magas száma miatt van: azt az esetet tekintettük FN találatnak, ha az ablak szerinti kézi annotációnál az *alapértelmezett* értéket kapta egy elem, a teljes mondat alapján azonban már specifikusabb eredményre jutottunk. Tehát az alulspecifikáltságot

DOI:10.15774/PPKE.ITK.2018.007

algoritmus	TP	FP	FN	pontosság	fedés	F-mérték
eredeti	1 590	37	761	97,73%	67,63%	79,94%
javított	2 103	37	248	98,27%	89,45%	93,65%

5.8. táblázat. Az ablak alapján történő kézi annotálás eredményeinek összehasonlítása a teljes mondatot figyelembe vevő kézi annotálás eredményeivel. (A „javított" sor a hibaanalízis utáni, javított algoritmus (a melléknevek és melléknévi igenevek *default*_0 helyett mindenhol 0) eredményeit tükrözi.)

tekintjük FN eredménynek (lásd az 5.9. táblázatot a FN találatok részletes eloszlásához).

hibatípus	hibaszám
FN	761
NOM helyett α	186
GEN helyett α	56
0 helyett *default*_0	519

5.9. táblázat. A kétféle kézi annotáció összehasonlításakor megfigyelt *álnegatív* eredmények. Az egyes sorok azt mutatják, milyen eset helyett milyen esetet egyértelműsített (alulspecifikálva) a csak az ablak alapján történő kézi annotáció.

Ugyanakkor látni kell azt is, hogy ezek nem feltétlenül hibák - a főnevek *alapértelmezett* esetének számító α pontosan a NOM és a GEN esetragot fogja össze: ezekben az esetekben az történik, hogy az ablak alapján még nem állapítható meg egyértelműen, hogy az α esetragú névszó alanya vagy egy birtokos szerepű tagja a mondatnak, ezt csak a tágabb kontextus segítségével lehet eldönteni. Fontos külön említeni azonban a *default*_0-k kiugróan magas számát a mellékneveknek és melléknévi igeneveknek esetén, amelyek az *alapértelmezett* esetet kapták a kézi annotáció során az ablak alapján, de az esetük a mondatot figyelembe véve 0. Összesen csak 6 olyan eset fordult elő, hogy az ablak alapján *default*_0-nak ítélt esetet a mondat teljes egésze NOM-ként egyértelműsítette, minden más alkalommal 0 lett ezekből. A (25a) példában szemléltethető, hogy az ablak alapján még elképzelhető, hogy az adott token a mondat alanya lesz, vagy egy jelöletlen

DOI:10.15774/PPKE.ITK.2018.007

birtokos szerkezetben a birtokos, de a teljes kontextus (25b) egyértelművé teszi, hogy ez egy főnevet módosító elem.

(25) a. *telepített mintegy negyven*
ablak alapján: telepített telepít IGE._MIB.*default*_0

b. *Kétéves koromban elvesztettem anyai nagyszüleimet, s velük együtt a szülőfalumból Magyarországra* ***telepített*** *mintegy negyven családban szinte minden rokonomat.*
mondat alapján: telepített telepít IGE._MIB.0

Az eredmények összefoglalásaképpen elmondható, hogy a magas pontosság megfelel az eljárással kapcsolatosan támasztott elvárásoknak. A fedés javítható (az így kapott eredményekhez lásd az 5.8. táblázatot), ha a melléknevek és melléknévi igenevek az *alapértelmezett* érték helyett automatikusan a 0 esetet kapják meg az ablak alapján.

5.8.3.2. Az algoritmus kiértékelése

Az 5.10. táblázatban az algoritmus teljesítményének kiértékelése látható. Ebben az esetben a gépi esetegyértelműsítést hasonlítottuk össze a csupán az ablakot figyelembe vevő kézi annotációval.

algoritmus	TP	FP	FN	pontosság	fedés	F-mérték
eredeti	2 220	63	125	97,24%	94,67%	95,94%
javított	2 332	63	13	97,37%	99,45%	98,40%

5.10. táblázat. Az algoritmus teljesítményének kiértékelése a csak az ablakot figyelembe vevő kézi annotáción. (A „javított” sor a hibaanalízis utáni, javított algoritmus (az `NPMod` kategóriájú elemekre *default*_0 helyett mindenhol 0) eredményeit tükrözi.)

Mind a pontosság (97,24%), mind a fedés (94,67%) kiemelkedően magas. Az algoritmus az ablak alapján egyértelműsíthető tokeneket egyértelműsíti, de nem specifikál olyan eseteket, amiket még nem lehetne. Az 5.11. táblázatban érdemes

DOI:10.15774/PPKE.ITK.2018.007

megfigyelni a hibák eloszlását, különös tekintettel utolsó sorára, ahol az algoritmus túlzottan alulspecifikál; *default* értéket illeszt a melléknevekre, melléknévi igenevekre, holott 0-t kéne. Ez párhuzamba hozható az 5.8. táblázatban látható, a 25a. és a (25b) példákban bemutatott jelenséggel. Az 5.11. táblázatbeli 112 eset tovább erősíti azt a hipotézist, hogy érdemes az `NPMod` kategóriájú elemekre *default_*0 helyett mindig 0-t illeszteni (az így kapott eredményekhez lásd az 5.10. táblázatot).

hibatípus	hibaszám
FN	125
NOM helyett α	9
GEN helyett α	4
0helyett *default_*0	112

5.11. táblázat. Az algoritmus teljesítményének kiértékelésekor megfigyelt *álnegatív* eredmények. Az egyes sorok azt mutatják, milyen eset helyett milyen esetet egyértelműsített (alulspecifikálva) az algoritmus.

A fentiek alapján látható, hogy a bemutatott algoritmus[1] az ismertetett eredmények alapján igen magas pontossággal és fedéssel teljesített. A további kutatási feladatok közé tartozik a vokatívusz eset feltérképezése és a jelenlegi algoritmus kiegészítése a vokatívusz kezelésével, illetve a többtagú nevek belsejében szereplő elemek esetének megvizsgálása. Ezeket az ablakban szereplő információk mellett a korábban már feldolgozott elemekre támaszkodva szükséges egyértelműsíteni. Mindezt pedig követheti az állítmány-esetű névszók detektálása a tágabb kontextus alapján.

5.8.4. A jelölt birtokos és kapcsolódó esetek

A jelölt birtokos esetén a birtokos és a datívuszi igei vonzatok között kell döntenünk. Az ilyen elemek mindenképpen bekerülnek a tározóba kínálatként, és egy igei elem vonzatkeresőjét vagy egy birtokos ragozású elem birtokoskeresőjét

[1] Az eljárás implementációja, a tesztfájl és az annotált fájl elérhető itt: `https://github.com/ppke-nlpg/nom-or-what`.

DOI:10.15774/PPKE.ITK.2018.007

elégíthetik ki; azt, amelyik előbb jön[1]. Ilyenkor előfordulhatnak olyan szerkezeti többértelműségek, amelyek az emberi mondatmegértés számára is csak a kontextus vagy a világismeret alapján oldhatóak fel[2]. A (26) mondat esetében kontextus nélkül kétféle jelentés is előhívható: a (26a) jelentés esetén az embereknek a segítsége egy -nAk ragos birtokost tartalmazó birtokos szerkezet, a *kell* ige datívuszi vonzata pedig nem hangzik el. A (26b) jelentés esetén az az embereknek a *kell* ige datívuszi vonzata, az *a segítsége* birtokosa pedig egy zéró névmás, amelynek koreferenciáját a kontextusból kellene kiszámítani.

(26) a. *Kellett* $\emptyset_{Dat}$ ***az embereknek a segítsége.***
a datívuszi vonzat néma

b. ***Kellett az embereknek*** *a segítsége* $\emptyset_{PersSg3}$.
a birtokos néma

Az ilyen szerkezetek messze meghaladják a gép által jelenleg kezelhető szintet, így többet jelenleg nem lehet mondani róluk.

5.8.5. A birtokos élek létrejötte

Egy kereslet-kínálat elvű rendszerben fontos eldönteni, hogy „ki keres kit”, azaz hogy a birtok vagy a birtokos viselkedjen-e kínálatként. Mivel a birtok minden esetben egyértelműen jelölve van a ragozása által, ezért ő kell, hogy indítsa a keresést. Az egyértelműsítésre váró birtokosok pedig kínálatként jelennek meg. A `Nom-or-What` keresőeljárás nem eredményezi semmilyen esetben a függőségi él létrejöttét, csak az őt indító elem esetének tisztázására szolgál, az NP egy megfelelő esetet és **CAS** jegyet kaphasson.

A birtok (a birtokos ragozást magán viselő elem) tehát kétféle lehetséges birtokost kereshet magának a mondatban: egy **Dat** főkategóriájú elemet vagy egy GEN jegyű elemet. A birtokoskereső eljárás működése a következő négy lépéssel írható le:

[1]Ez az elemzőrendszer jelenlegi heurisztikája. A jelenség pontosabb megértéséhez további kutatás szükséges.

[2]Habár az emberi elemző számára ezek a szerkezetek a legtöbb esetben egyértelműnek tűnnek, és csak sokadik olvasásra veszi észre a „kerti ösvényt”.

DOI:10.15774/PPKE.ITK.2018.007

1. a birtokos számának és személyének kiszámítása a ragozásból

2. GEN jegyű, egyező elem keresése balra a tározóban
   HA talált → birtokos él
   KÜLÖNBEN

3. **Dat** főkategóriájú, egyező elem keresése balra a tározóban
   HA talált → birtokos él
   KÜLÖNBEN

4. **Dat** főkategóriájú, egyező elem keresése jobbra az ablak jobb szélétől

Az 1. lépésben az elemző a birtokos számát és személyét kiszámítja a ragozásból, majd létrehoz egy zéró csomópontot, ami tartalmazza a megfelelő számot és személyt. Ez a csomópont addig tölti be a birtokos funkcióját, ameddig nem érkezik egy illeszkedő testes jelölt.

A 2. és 3. lépésben a megfelelő jegyű és főkategóriájú testes birtokos keresése történik. Találat esetén létrejön egy ideiglenes birtokos él. A tagmondat végéig ez az él felbontható, a tagmondat végén pedig véglegessé válik. Ha balra nem talált megfelelő jelöltet, a tározóba bekerül a **Dat** főkategóriájú elemet kereső birtokoskereső. Vegyük észre, hogy már a birtok elemzési ablakában lehet egy **Dat** főkategóriájú elem, de mivel a -nAk ragos birtokos szerkezet birtokosa és birtoka két külön fráziskénti kezelődik, ezért ez a birtokos kapcsolat később jön csak létre.

A fentiekhez hasonlóan az elemző képes kezelni a többszörös birtokosokat is. Az ilyen a mondatoknál is a fenti lépéseket végzi el, azzal a különbséggel, hogy minden elemre balról jobbra haladva történik az elemzés csak úgy mint a beágyazás nélküli birtokos szerkezetet tartalmazó mondatoknál: az NP-k előkészítésekor a testes esetrag nélküli elem esetegyértelműsítése után a birtokok megkeresik a saját birtokosukat, és a mondat végére érve minden függőségi él létrejön.

DOI:10.15774/PPKE.ITK.2018.007

5.9. Az igék vonzatkeretének egyértelműsítése

5.9.1. Az infinitívuszi vonzat és az igekötő viszonya

Az 5.7.2. fejezetben ismertetett korpuszmérések eredményei mellett elméleti nyelvészeti oldalról is megközelítettük az igekötők és infinitívuszi vonzatok vonzategyértelműsítő szerepének vizsgálatát. Öt igeosztályba soroltuk az igéket aszerint, hogy igekötő nélküli, illetve igekötős vonzatkeretükben szerepelhet-e infinitívuszi vonzat. Az 5.12. táblázat foglalja össze az öt igeosztály tulajdonságait.

	igeosztály	példa		
	PreV vagy INF vonzat	tő	PreV	INF
(a)	nincs PreV, nincs INF	*villog*	X	X
(b)	nincs INF	*esik*	el, le...	X
(c)	nincs PreV	*kell*	X	?
(d)	PreV és INF kölcsönösen kizárják egymást	*tud*	le, meg...	?
(e)	INF bizonyos PreV-vel	*megy*	ki, el...	?

5.12. táblázat. Öt igeosztály (Vadász, Kalivoda és Indig 2017) az igéknek az igekötővel és az infinitívuszi vonzattal való kombinálhatósága alapján (X: nincs elfogadott infinitívuszi vonzat vagy igekötő az igével kombinálva, ?: az igének lehet, hogy van infinitívuszi vonzata)

A balról jobbra elemzés szempontjából azt tekintjük ideális állapotnak, ha az ige elemzésének pillanatában minden információval rendelkezünk, amely a vonzatkeret egyértelműsítéséhez szükséges. Ezért megvizsgáltuk az igekötő (PreV), finit ige (FIN) és az infinitívusz (INF) egymáshoz viszonyított sorrendjének összes konfigurációját, melyet az 5.13. táblázat foglal össze.

A leggyakoribb szerkezet az olyan PreV–FIN–INF, amelynél az igekötő az infinitívuszhoz tartozik. Szintén gyakori a FIN–PreV–INF sorrend, ekkor az igekötő legtöbbször a finit igéhez tartozik. Ez a szórend jellemzi a non-neutrális (ezen belül főként a felszólító, tagadó) mondatokat. Az INF–FIN–PreV szerkezetnél az figyelhető meg, hogy maga az infinitívusz áll a fókuszpozícióban, és ez mozdítja el az igekötőt a közvetlenül preverbális pozícióból. A további három lehetséges

DOI:10.15774/PPKE.ITK.2018.007

PreV – **FIN** – INF	**meg** sem **próbálták** csökkenteni
PreV – FIN – **INF**	**le** is akartam **fényképezni**
FIN – **PreV** – INF	**szűnjön meg** létezni
FIN – **PreV** – **INF**	sikerült két példányt **el** is **ejtenie**
INF – **FIN** – **PreV**	csodálni **járok vissza**
INF – FIN – **PreV**	**rohannia** kell **vissza**
FIN – INF – **PreV**	-
FIN – **INF** – **PreV**	kellett egészben új állami rendet [...] **építeni fel**
INF – **PreV** – **FIN**	kártyázni **le** ne **ülj**
INF – **PreV** – FIN	**feledni el** nem tudlak
PreV – INF – **FIN**	-
PreV – **INF** – FIN	**el** nem **utasítani** kegyeskedjék

5.13. táblázat. A finit ige (FIN), az infinitívuszi vonzat (INF) és valamelyik igekötőjének (PreV) egymáshoz viszonyított lehetséges sorrendjei példákkal (vastag betűvel az összetartozó párokat jelöltük) (Vadász, Kalivoda és Indig 2017)

kombinációra is található példa a korpusz mondatai között (lásd az 5.13. táblázat utolsó három sora), ezek a szerkezetek azonban ritkák.

5.9.2. A VFrame eljárás

Az eddigiekből tehát látható, hogy az igei elemet megelőző összes, és az azt követő néhány token ismerete egyértelműsíti a vonzatkeretet az igekötő és az infinitívuszi vonzat tekintetében. Lehetséges, hogy még így is többértelműség áll fenn, de a többértelműségek nagy része kiszűrhető az igekötő és az infinitívuszi vonzat ismeretében. A VFrame egy olyan keresőeljárás, mely minden igei elem összes igekötő–infinitívuszi vonzat kombinációját kezeli (beleértve azokat az eseteket, amikor nincs igekötő és/vagy infinitívuszi vonzat). A VFrame szerkezetét az 5.3. ábra mutatja, mely megfelel az 5.4. fejezetben bemutatott általános keresőeljárás egy speciális változatának.

Az **Igekötő** jegy az igei elemmel kompatibilis *összes lehetséges igekötő halmazát* tartalmazza (függetlenül az infinitívuszi vonzattal való viszonyától), vagy X-et (ha az adott igei elemnek semmilyen igekötője nem lehet). Ez a tulajdonság töltődik fel az igei elem az aktuális mondatban *megtalált igekötőjével*[1]. Az **Infini-**

[1]Abban az esetben, ha az igekötő az igén van, az elemző kihagyja a VFrame keresőeljárást,

DOI:10.15774/PPKE.ITK.2018.007

$$\text{VFrame}\begin{bmatrix}\text{Irány} = > \mid < \\ \text{Igekötő} = \textit{lehetséges igekötők halmaza} \mid X \mid \textit{talált token} \\ \text{Infinitívusz} = ? \mid X \mid \textit{talált} \\ \text{Találati függvény} = \textit{találatkor vagy a sikertelen keresés végén fut le} \\ \text{Egyéb}\begin{bmatrix}\text{Tő} = \textit{az ige töve} \\ \text{megszorítási függvény} = \textit{a találatok megszorítási szabályai}\end{bmatrix}\end{bmatrix}$$

5.3. ábra. A *VFrame* keresőeljárás architektúrája (Indig és Vadász 2016b)

tívusz jegy jelzi, hogy az igei elemnek lehet-e infinitívuszi vonzata (*?*) vagy sem (*X*). Ha az eljárás talál infinitívuszi vonzatot, akkor azt az Infinitívusz jegy *Talált* állása jelzi, amely egyúttal az infinitívusz tokenjére mutat. A **Találati függvény** tartalmazza azt a függvényt, amelyet az elemző egy elem (igekötő vagy infinitívuszi vonzat) megtalálásakor, vagy annak hiányában meghív, és kezeli a különböző keresőeljárások állapotai közötti átmeneteket is (lásd a D.1. ábra). A **Megszorítási függvény** kezeli a találat megszorításait, mely frissíti a keresendő elemek halmazát, például ha az INF és a PreV kölcsönösen kizárják egymást.

A VFrame, több keresőeljárás sorozatával – melyek viszonyai egy három valódi állapottal rendelkező, véges állapotú automata segítségével írhatóak le (lásd a D.1. ábra) – képes arra, hogy a megfelelő keresőeljárások elindításával egyértelműsítse az aktuális vonzatkeretet[1]. Ha a mondatban több olyan elem is van, amelyhez tartozhat igekötő – és ezek lehetséges igekötői között átfedés van –, a VFrame keresőeljárással az igék, igekötők és infinitívuszi vonzatok akkor is helyesen kapcsolhatók össze, ha ez nem kombinálódik más problémával. Az olyan példákban, ahol egynél több infinitívusz jelenik meg az igei komplexumban, az infinitívuszok jellemzően egymás mellett állnak, pl. *el kell kezdeni keringőzni tanulni*, *el fogod tudni dönteni*. Arra is vannak példák, hogy az egyik infinitívusz az igei komplexum élére kerül, pl. *pisilni el tudtál menni*. Ilyenkor minden igei elem, így a főige és a mondatban szereplő infinitívuszi vonzatok egyaránt elindítják a saját VFrame keresőeljárásukat a megfelelő beállításokkal. Az olyan mondatok

és elindítja a vonzatok keresését.

[1]A VFrame pontos működéséről és implementációjáról lásd (Indig és Vadász 2016b; Vadász, Kalivoda és Indig 2018).

DOI:10.15774/PPKE.ITK.2018.007

esetén, amelyeket a vonzatok nem természetes sorrendje miatt az ember is nehezen elemez, az elemző visszalépéssel és újraelemzéssel alakítja ki a megfelelő igei elem–igekötő és igei elem–infinitívuszi vonzat kapcsolatokat.

5.9.3. A A VFrame eljárás szótára

A VFrame eljárás működésének alapfeltétele egy szótár, amelyben az igékhez rendelt igekötők tárolódnak azzal az információval együtt, hogy az ige-igekötő pár vonzatkeretei között szerepel-e olyan, amely infinitívuszi vonzatot tartalmaz. Ez a szótár a Manócska (lásd a 4.2.2. fejezet) felhasználásval készült, pontosabban annak az ige-igekötő és az ige-infinitívusz listájából. Az 5.4. ábra két bejegyzést mutat ebből a szótárból[1].

```
utál {?, el:X, ki:X, meg:X}
felejt {?, el:?, ki:X, le:X, ott:X, rajta:X}
```

5.4. ábra. Két bejegyzés a VFrame által használt szótárból. Az igekötőhöz rendelt *?* azt jelenti, hogy azzal az igekötővel együtt az igének lehet infinitívuszi vonzata, az *X* azt jelenti, hogy nem. Az igéhez rendelt *?* azt jelenti, hogy az igének igekötő nélkül lehet infinitívuszi vonzata.

A szótár segítségével ki lehet értékelni az eljárás teljesítményét, melyet a következő fejezetben ismertetek.

5.9.4. A A VFrame eljárás kiértékelése

A VFrame teljesítményét 1 000 tesztmondaton mértük ki[2]. A tesztmondatokat (egész pontosan a tagmondatokat, amelyekben finit ige szerepel) az MNSZ 2.0.4 szolgáltatta. A VFrame teljesítményét három jelenség kezelése teszi ki: 1) a finit ige és az igekötőjének összekapcsolása, 2) az infinitívusz és az igekötőjének összekapcsolása, valamint 3) a finit ige és az infinitívuszi vonzatának összekapcsolása.

[1] Az 5.4. ábrán szereplő *megutál* igének elvileg lehet infinitívuszi vonzata, de az MNSZ 2.0.4 korpuszban mindössze egy példa volt erre: *már megutáltam folyton hasznos lenni.* Az ebből előállított gyakorisági lista ötös gyakoriságnál kezdődik, így néhány ritka eset nem került be az erőforrásba.

[2] Az implementációnk elérhetősége: `https://github.com/ppke-nlpg/vframe`

DOI:10.15774/PPKE.ITK.2018.007

A kiértékelés során a finit igékre koncentráltunk, a többi igének (így a melléknévi és határozói igeneveknek) és az igekötőjének vagy infinitívuszi vonzatának az összekapcsolását kihagytuk a vizsgálatból.

A tesztmondatokat ennek megfelelően úgy válogattuk, hogy egy finit igét tartalmazzanak, ezen kívül a tagmondatban legyen legalább vagy egy igekötő, vagy egy infinitívusz. A finit ige és az infinitívusz is lehet igekötős, hiszen a VFrame ezt külön esetként kezeli. Az így leszűrt mondatok közül véletlenszerűen választott 1 000 darabot vettünk fel a teszthalmazba. A tesztmondatok, valamint az összetételükkel kapcsolatos részletes információ megtalálható a VFrame git repozitóriumban[1].

A mondatokhoz kézzel megjelöltük a bennük található ige-igekötő, infinitívusz-igekötő, ige-infinitívusz kapcsolatokat, amely referenciaadatként szolgált a kiértékeléshez. A kézi annotációt és a VFrame kimenetét automatikusan összevetettük, és a megegyező vagy különböző eredményeket a megfelelő kategóriákba soroltuk, amelyeket az 5.14. táblázat tartalmaz.

kategória	finit ige/infinitívusz–igekötő	finit ige–infinitívusz
TP	van igekötő és megtalálta	van infinitívusz és megtalálta
TN	nincs igekötő és nem találta meg	nincs infinitívusz és nem találta meg
FP	rossz igekötőt talált	rossz infinitívuszt talált
FN	nem találta meg az igekötőt	nem találta meg az infinitívuszt

5.14. táblázat. Az egyes kategóriák, amelyek az igekötő-ige és az ige-infinitívusz összekapcsolásánál felmerülnek. **TP**: valós pozitív, **TN**: valós negatív, **FP**: álpozitív és **FN**: álnegatív

A kategóriák számosságát mindhárom feladatra külön megnéztük, így megvizsgálható a VFrame teljesítménye az ige-igekötő, az infinitívusz-igekötő, a finit ige/infinitívusz-igekötő, valamint az ige-infinitívusz összekapcsolására is, de a VFrame teljesítményére összességében is. A VFrame teljesítményét összevetettük két egyéb eljárásával is. Az eredményeket az 5.15. táblázat mutatja.

[1] `https://github.com/ppke-nlpg/vframe`

DOI:10.15774/PPKE.ITK.2018.007

A VFrame teljesítményét összevetettük egy Recski Gábor által javasolt (Recski 2011), nagyon egyszerű heurisztikán alapuló eljárás[1] apró módosításával, azaz az algoritmus ahelyett, hogy az igekötőt keresné az igéhez, minden igekötőhöz a hozzá legközelebb álló igét (finit igét vagy infinitívuszt) rendeli. Hasonlóan jár el az infinitívuszok esetén is, melyekhez egy finit igét keres[2], nem pedig fordítva. Ezt az eljárást tekintettük alapvonalnak (a továbbiakban baseline néven hivatkozom rá), mivel nem támaszkodik arra a szótáron alapuló információra, hogy az igének lehet-e infinitívuszi vonzata, illetve hogy milyen igekötője lehet egyáltalán. Csupán annyi megszorítással él, hogy bizonyos finit igéknek nem keres igekötőt (ezek a segédige-szerű igék (Kálmán C. et al. 1989) alapján az *akar, bír, fog, kell, kezd, kíván, lehet, mer, óhajt, próbál, szabad, szándékozik, szeret, szokik, talál, tetszik* és a *tud*, a létigével kiegészítve).

A baseline módszer mellett a `magyarlanc` függőségi elemzőjének (Zsibrita, Vincze és Farkas 2013) eredményével is összevetettük a VFrame teljesítményét. A függőségi elemzésben megnéztük, hogy hányszor egyezett meg a kézi annotációval az ige-igekötő és az ige-infinitívusz összekapcsolása. Ez az összekapcsolás gyakran amiatt volt hibás, hogy az elemző eleve rosszul állapította meg a finit igét (összesen 40 alkalommal).

Az eredmények azt mutatják, hogy a VFrame és a baseline módszer teljesítménye között csupán kis különbség van. A baseline módszer néhány alfeladatban a fedés szempontjából valamivel jobban teljesített, míg a VFrame minden alfeladatban és összesítve is a pontosságban volt jobb. A baseline módszer a tesztmondatok korpuszból származó szófaji címkéjére támaszkodik, így előfordult, hogy nem helyesen állapította meg a finit igét (pl. a *vagy* kötőszót vette finit igének). Ebből a hibából összesen 4 darab fordult elő. A baseline módszerhez képest a VFrame a felhasznált szótárnak köszönhetően tudott jobban teljesíteni, amely segítségével kizárhatóak a helytelen igekötő-ige vagy infinitívusz-ige kapcsolatok.

A baseline módszer és a VFrame esetében a két hibatípusba (FP, FN) tartozó hibákat megvizsgálva kiderül, hogy a legtöbbjük az eleve hibás bemenetből

[1] Az eljárás a finit igét és az infinitívuszt az igekötővel, valamint a finit igét az infinitívuszi vonzatával azok közelsége alapján kapcsolja össze.

[2] Mind az igekötő, mind az infinitívusz összekapcsolásának feltétele, hogy egy tagmondatban szerepeljenek a finit igével, ez a feltétel a tesztmondatainkban mindig teljesül.

DOI:10.15774/PPKE.ITK.2018.007

		Fin–Ik	Inf–Ik	Fin/Inf–Ik	Fin–Inf	Összesen
pontosság	VFrame	**97,57**	**94,71**	**96,82**	**97,88**	**97,21**
	baseline	92,39	90,40	91,87	96,98	93,72
	`magyarlánc`	88,22	89,36	88,53	89,93	89,08
fedés	VFrame	96,30	**94,21**	**95,76**	98,34	96,70
	baseline	**96,49**	92,75	95,50	**99,05**	**96,80**
	`magyarlánc`	79,20	86,15	80,96	89,74	84,23
F-mérték	VFrame	**96,93**	**94,46**	**96,29**	**98,11**	**96,95**
	baseline	94,40	91,56	93,65	98,00	95,24
	`magyarlánc`	83,47	87,73	84,58	89,83	86,59

5.15. táblázat. A különböző alfeladatok és a VFrame teljesítményének kiértékelése, összevetve egy baseline eljárással és a `magyarlanc` függőségi elemző eredményével. **Vastag betűvel** szedtük a legmagasabb értékeket.

fakad. Mindkét eljárás esetében a korpuszból vett tesztmondatokban a szófajegyértelműsítő hibát vétett (például az *elég* főnevet igekötős finit igének jelölte meg).

Egy másik, hibát okozó jelenség az, amikor a példamondatban az ige töve nem a megfelelő módon van feltüntetve – elsősorban az ikes igék esetében. Például a *mit lélegeznek ki a falak* tesztmondatban a *lélegeznek* ige töve a korpuszban a következő formában jelenik meg: *lélegezik/lélegzik*. Az ige-igekötő-infinitívusz listában azonban ez a tő nem szerepel (hiába van felsorolva a *ki* igekötő a *lélegez* tőhöz a szótárban). A VFrame esetében más, relevánsabb hibatípust nem találtunk, tehát a hibás eredmény lényegében a hibás bemenetből adódik.

A `magyarlanc` eredménye mind a baseline módszerrel, mind a VFrame módszerrel szemben alulmaradt. A hibák számát jelentősen növelte, hogy a másik két módszerhez képest többször rontotta el a finit ige megtalálását.

Mindent összevetve a VFrame előnye elsősorban abban áll – a legmagasabb pontosság és F-mérték mellett –, hogy a balról jobbra és szavanként történő feldolgozás miatt beépíthető az AnaGramma elemzőbe. Mivel a baseline és a VFrame algoritmus nem mond ellent egymásnak, így a szótárban nem szereplő igék esetén tartalék eljárásként használható, kiküszöbölve a szótár gyengeségeit.

DOI:10.15774/PPKE.ITK.2018.007

5.9.5. Megoldatlan nyelvi jelenségek

A VFrame eljárás a homonímia miatt hibásan címkézett igekötőket jelenleg nem kezeli. A *meg* gyakran igekötő címkét kap akkor is, ha mellérendelő kötőszó, a *ki* igekötő pedig gyakran keveredik az azonos alakú vonatkozó illetve kérdő névmással. A hibásan annotált szavak utólagos, automatikus javítását nehezítik azok az esetek, amikor ezek valóban létező kombinációt alkothatnának az igével és olyan pozícióban állnak, amely az igekötőé is lehetne (lásd a (27b) példát).

(27) a. akkor csak lámpát kell vennem **meg** rácsot
a *meg* igekötőként a létező *meg+vesz* igét eredményezheti
b. az mennyibe fog kerülni és **ki** fogja rá adni a pénzt
a *ki* igekötőként a *ki+ad* létező igét eredményezheti

A korpuszból kinyert mondatokban több mint 200 példát találtunk egy különleges szerkezetre, amelyben látszólag nem tartozik ige az igekötőhöz. A szerkezet egy infinitívuszból, egy finit igéből (jellemzően segédigéből) és egy olyan igekötőből áll, amely az infinitívuszon is megjelenő igekötő hangsúlyos alakja. Például: ***elképzelni*** *bármit* ***el*** *lehet,* ***becsajozni be*** *tudnék,* ***megírni meg*** *kell.* Ennek a szerkezetnek a kezelése még nem megoldott, de ritkasága miatt nem is elsődleges prioritás. A particípiumok ugyan rendelkeznek vonzatkerettel, de – hasonlóan az igekötőkhöz – az annotációval kapcsolatos problémákat a VFrame eljárás nem tudja megoldani. Például ilyen a befejezett melléknévi igenév–melléknév–múlt idejű ige szófaji többértelműség kezelése, mely további kutatás tárgyát képezi. Az ilyen igei elemeknél a VFrame eljárást megszorítjuk, hogy az igekötőt vagy az infinitívuszi elemet csak az igenevet tartalmazó NP határain belül és csak balra (a tározóban) keresse.

DOI:10.15774/PPKE.ITK.2018.007

5.10. Összefoglalás és kapcsolódó tézisek

A fejezetben bemutattam az AnaGramma elemző működését a megvalósítás szempontjából, valamint a nyelvi jelenségek kezeléshez használt ablakot, melynek ötlete a két fázisban működő *Sausage Machine*-ből származik. Az általam bemutatott ablak – mely a *Sausage Machine* első, PPP fázisának felel meg – és a rajta definiált keresőeljárások megoldást adnak a hatékony, emberi elemzőhöz hasonló balról jobbra elemzés számos problémájára.

9. Tézis. *Létrehoztam egy új megközelítésű (az ún. ablakra épülő) elemzési modell alapjait, amelynek elméletét társszerzővel közösen dolgoztam ki, és melynek segítségével a magyar nyelvű bemenet hatékonyan és az emberi feldolgozáshoz hasonlóan, szigorúan balról jobbra feldolgozható.*

A tézist alátámasztó közlemények: [14, 5, 21, 34, 25, 26]

A definiált ablakon működő eljárások közül ismertettem a jelöletlen (nominatívuszos) névszók egyértelműsítését és a jelölt (-nAk ragos) birtokos szerkezetek (Bánréti et al. 1992) kezelését. A módszerben a kétfázisú mondatelemzés első fázisában az előretekintő elemzési ablak segítségével megtörténik a testes esetrag nélküli elemek eset-egyértelműsítése, melynek eredményeképpen tisztázódik a mondatbeli szerepük. Birtokos esetén, a bemutatott kereslet-kínálat keretrendszerben a birtok lesz az, amely birtokos-kereslettel él, amely kereslet kielégülésekor birtokos él jön létre a birtok és birtokosa között.

10. Tézis. *Létrehoztam egy az ablak segítségével a jelöletlen szerkezetek egyértelműsítését (például a magyar birtokos szerkezet és az alanyeset hatékony, valós idejű elkülönítését) elvégző algoritmust, melynek elméletét társszerzővel közösen közösen dolgoztam ki.*

A tézist alátámasztó közlemények: [5, 21, 25]

Korpuszmérések alapján bizonyítottam, hogy az AnaGramma elemzőrendszer keretein belül a finit ige–igekötő kapcsolat létrehozása mellett (**Indig** és Vadász 2016b) az infinitívusz–igekötő és a finit ige–infinitívuszi vonzat kapcsolatok

DOI:10.15774/PPKE.ITK.2018.007

létrehozásához is elegendő a feltételezett két token méretű elemzési ablak használata. A tározó és az ablak segítségével a VFrame keresőeljárás a mondatban szereplő igei elemeket (finit és infinit igéket), valamint az igekötőket a megfelelő módon kapcsolja össze.

Az aktuális finit ige–igekötő–infinitívuszi vonzat kapcsolatok létrejötte után elindulnak a megfelelő vonzatkeresők, amelyek mind a tározóban, mind a mondat hátralévő részében keresik a vonzatkeret elemeit. Amennyiben a VFrame nem egyértelműsíti teljesen a vonzatkeretet (mert egy ige ugyanazzal a finit ige–igekötő–infinitívuszi vonzat viszonnyal többféle vonzatkerettel is rendelkezhet), akkor az összes ennek megfelelő vonzatkeret vonzatkeresője elindul. Ekkor a mondatban aktuálisan szereplő többi vonzat egyértelműsíti a vonzatkeretet.

11. Tézis. *Létrehoztam a VFrame eljárást, amelynek elméletét társszerzővel közösen dolgoztam ki, amivel a magyar nyelvben a helyes igekötő megtalálása az igekötők eloszlási mintájának ismeretében a lehetséges igei vonzatkeretek halmazának leszűkítésével történik.*

A tézist alátámasztó közlemények: [14, 26, 27]

Az eredményeim sokrétűen alkalmazhatók a magyar szövegek elemzésében. A különböző mondatok a bemutatott módszerekkel történő elemzés után leegyszerűsíthetők mondatvázakká, amely megkönnyíti a további szerkezetek kutatását, melyek eredményeképpen egy a teljes nyelvet leíró, az emberi elemzés működését figyelembe vevő elemzőrendszer jöhet létre. Az emberi elemzőt vizsgáló pszicholingvisztikai kutatások számára az általam definiált modell támpontot nyújthat, mely a későbbi kutatásokat segíti. A maximális főnévi csoportok pontosabb meghatározása az esetegyértelműsítés által a felszíni elemzést, illetve az információ visszakeresést is javítani képes.

DOI:10.15774/PPKE.ITK.2018.007

6. fejezet

Az új tudományos eredmények összefoglalása

„Mindörökre felelős vagy azért, amit egyszer megszelídítettél."

(Antoine de Saint-Exupéry:
A kis herceg,
Takács M. József fordítása)

A dolgozatban bemutattam a magyar nyelvre jelenleg is használt nyelvtechnológiai szerelőszalag működését. A szerelőszalag-architektúra számtalan előnnyel és hátránnyal rendelkezik. Napjainkra a régóta ismert előnyök mellett lassan a hátrányok is megmutatkoznak. Az egyes modulok csak a szomszédos modullal érintkeznek, így a bemenetük és kimenetük nagyban eltérhet. Manapság több eszköz is elérhető egy adott feladat megoldására, ezért szükségessé vált azok egységesítése, együttműködésük vizsgálata, az ökoszisztémáik működéséhez szükséges feltételek kialakítása.

Ismertettem az általam fejlesztett, pszicholingvisztikailag motivált nyelvelemző modellel, az AnaGramma elemzővel szemben támasztott elvárásokat, melynek célja egy emberi elemzőhöz hasonlító számítógépes szövegelemzési modell létrehozása. Továbbá, hogy megszüntesse a soros architektúrából származó hibákat, melyek a szerelőszalag végére felerősödnek, és értékelhetetlenné teszik az eredményt. Az általam készített elemzőrendszer eredendően párhuzamos, így minden modul egyszerre, egymást javítva képes futni benne, azaz nem támaszkodnak feltétlenül egymás eredményeire. A dolgozat további részében a modell

DOI:10.15774/PPKE.ITK.2018.007

architektúrájához szükséges eljárásokat tekintettem át, melyekből ötletet merítek az elemző moduljainak elkészítéséhez.

A dolgozatomban bemutatott téziseket négy csoportba lehet osztani. A főnévi csoportok keresésének state-of-the-art megoldásától módszeresen a szekvenciális címkézés különböző közös tulajdonságain át az n-gram modellek vizsgálatával eljutottam az elemzőhöz szükséges korpuszminták újelvű alkalmazásához. Ezt követően a főnévi csoportok igei argumentumként történő azonosításának vizsgálatakor a létező erőforrások összekapcsolásával és nyelvfüggetlen információk átvitelével a finom osztályozások módját vizsgáltam, melynek segítségével pontosítani lehet az eljárásokat. Végül az elemző architektúrájának ismertetésével összefüggésben bemutattam két feltérképezett és kezelt nyelvi jelenséget, melyek mintájára a többi jelenség is kezelhető.

I. téziscsoport

Az első téziscsoportban a főnévi csoportok keresésére koncentráltam. Magyar és angol nyelven vizsgáltam meg a jelenleg használt state-of-the-art módszereket, hogy megértsem, miként lehetne őket felhasználni az elemző működéséhez. Először az angol nyelvű state-of-the-art megoldást (Shen és Sarkar 2005) akartam adaptálni a magyar nyelvre. A módszer reprodukálása során azonban kiderült, hogy az `IOB` reprezentációk szavazásával elérhető nyereség csak mérési hiba eredménye és műtermék. Így az angol nyelven legjobb módszer nem volt alkalmazható magyar nyelvre, hiszen a vele elért eredmény nem valós.

1. Tézis. *Méréssel kimutattam, hogy nem helytálló az a szakirodalomból ismert állítás, amely szerint a különböző `IOB`-reprezentációk közötti szavazás szignifikáns javulást hoz az angol nyelvű főnévi csoportok meghatározásának minőségén.*

A tézist alátámasztó közlemények: [3]

Habár az angol nyelvű eredmények mélyebb betekintést engednek a szekvenciális címkézés ezen alkalmazásának működésébe, önállóan még nem voltak alkalmasak a magyar nyelvű főnévi csoportok keresésének további javítására. A magyar nyelvű state-of-the-art módszer (Recski és Varga 2012) vizsgálata során

DOI:10.15774/PPKE.ITK.2018.007

a maximális NP-k keresésének feladatában viszont felismertem, hogy a state-of-the-art módszer csak bigram címkeátmenet-modellt használ, mert a névelemfelismerésből jövő módszerből származik, és trigram címkeátmenet-modell használatával javítottam modell eredményén.

2. Tézis. *Az általam kifejlesztett HunTag3 program segítségével méréssel igazoltam (társszerzővel közösen), hogy a trigrammok használatával javulás érhető el a bigrammokhoz képest a magyar nyelvű maximális főnévi csoportok meghatározásában.*

A tézist alátámasztó közlemények: [8]

A javított magyar eljárás a maximális NP-k keresésének feladatában az eddiginél jobb eredményt hozott, de nem változtatott jelentősen a rendszer működésén. Ez a tény megkönnyíti a jövőbeli alkalmazhatóságát. A gyors, információ visszakereséshez használt sekély elemzők betanításánál az eredményeimet tehát érdemes figyelembe venni. Bár ezen rendszerek elsődleges célja a valódi elemzők helyettesítése és pontosabb információk szerzése a főnévi csoportok működéséről, nem lebecsülendő a jövőbeni szerepük, mivel az angollal ellentétben magyar nyelvre még mindig nem érhető el szabadon annyi jó minőségű és gyors magyar nyelvi elemző.

II. téziscsoport

A második téziscsoportban a főnévi csoportok keresésének feladatán elindulva felismertem, hogy a szekvenciális címkézési feladatok sok tulajdonságukban különböznek ugyan, de sokban hasonlítanak egymáshoz, továbbá az emberi elemzőhöz hasonlóan balról jobbra haladva működnek. Ez a megfigyelés segítségemre volt az elemző architektúrájának tervezésében, hiszen ezek a módszerek az emberi elemzőhöz hasonlóan a szövegen balról jobbra haladva működnek. Ezért a szekvenciális címkézés feladatain általánosan alkalmazható módszereken kezdtem dolgozni, melyet a már meglévő eredményeim javítására használtam. Ehhez kapcsolódóan a dolgozatban bemutattam az általam vizsgált lexikalizációs eljárások működését és hatását.

DOI:10.15774/PPKE.ITK.2018.007

3. Tézis. *Létrehoztam egy új, általános, szekvenciális címkézésre alkalmazható lexikalizációs eljárást, melynek első konkrét alkalmazása tetszőleges részszerkezetek hatékony azonosítását szolgálja.*

A tézist alátámasztó közlemények: [2, 3]

Az általam feltalált lexikalizációs eljárással és az optimális küszöbérték meghatározásával és alkalmazásával meghaladtam az angol nyelvű közvetlen összetevős keresés feladatán a state-of-the-art módszer teljesítményét.

4. Tézis. *Az általam kidolgozott eljárás angol nyelvű főnévi csoportokra méréssel igazolhatóan felülmúlja a jelenleg ismert módszerek F-mértékét.*

A tézist alátámasztó közlemények: [2, 3]

A mérések során megfigyeltem, hogy az `IOB` reprezentációk közötti konverziók különféle manipulációjával elérhető javulásnak milyen további peremfeltételei vannak. Továbbá, hogy mennyire fontos a megfelelően felkészített konverter alkalmazása, valamint az, hogy a címkéző program fenn tudja tartani a jólformáltságot a kimeneti címkesorozatok zárójelezésében. Ennek mérésére kidolgoztam egy metrikát, amit gyakorlatban alkalmaztam az angol nyelvű közvetlen összetevők keresésének feladatán.

5. Tézis. *Kidolgoztam egy zárójelezési módszert, mely egyfajta metrikaként a címkézési feladatra készített módszereket minőség szerint rendezni tudja.*

A tézist alátámasztó közlemények: [2, 3]

III. téziscsoport

Mivel a maximális főnévi csoportok az igék argumentumaiként funkcionálnak a mondatban, megvizsgáltam a rendelkezésre álló magyar nyelvű igei erőforrásokat (Indig, Vadász és Kalivoda 2017; Kalivoda 2016; Kornai, Nemeskey és Recski 2016; Sass 2015; Sass et al. 2010). A vizsgálat során arra jutottam, hogy egyikben sincs szemantikai információ a szintaktikai mellett, aminek segítségével tovább lehetne finomítani a főnévi csoportok osztályozását. Bemutattam a *Linked*

DOI:10.15774/PPKE.ITK.2018.007

Data[1] fogalmát, és a módszer erőforrásokra vonatkoztatott változatának ismertetése után bemutattam néhány angol nyelvű példát az összekapcsolt erőforrásokra (Prószéky, Mihaltz és Kuti 2013; Vossen et al. 1998). Majd ezen a vonalon elindulva a kétnyelvű magyar-angol MetaMorpho adatbázis (Prószéky, Tihanyi és Ugray 2004) és az angol VerbIndex (Loper, Yi és Palmer 2007) összekapcsolását tűztem ki célul azért, hogy nyelvfüggetlen szemantikai annotációt tudjak automatikusan átvinni az információban jóval gazdagabb VerbIndexből a MetaMorpho adatbázisba.

6. Tézis. *Létrehoztam egy automatikus módszert az 1-, 2- és 3-vonzatú igék magyar–angol vonzatkeretpárjainak összekapcsolására, melynek eredményeképpen sikerült angolról magyarra átvinni a megfelelő tematikus szerepeket.*

A tézist alátámasztó közlemények: [11, 12, 4, 22]

Az összekapcsolás részeként harmonizálni kellett a két erőforrás között az elemek megszorításait leíró ontológiákat, melyek között egy áthidaló fogalmakat tartalmazó ontológiával teremtettem meg az átjárhatóságot.

7. Tézis. *Kialakítottam egy ontológiát, amely összekapcsolja a magyar nyelvű MetaMorpho igéinek leírását az angol VerbIndex szintaktikai és szemantikus kategóriáival.*

A tézist alátámasztó közlemények: [11, 12, 4]

A meglévő információk alapján össze lehetett kapcsolni a magyar és az angol nyelvű WordNeteket is. Ezeket a kapcsolatokat is latba vetettem, hogy javítsam a minőséget, de azok nem bizonyultak megfelelőnek a feladat szempontjából.

8. Tézis. *Méréssel kimutattam, hogy a magyar és angol nyelvű WordNetek bevonásával nem lehet a fenti ontológia minőségét tovább javítani.*

A tézist alátámasztó közlemények: [11, 12, 4]

Végül az igei vonzatkeretek egy viszonylag jó fedésű alosztályára sikerült jó minőségben szemantikai információt átvinni automatikus úton, mely további osztályozási lehetőségeket nyitott meg. A szerzett tapasztalatok egyedülállóan előremutatóak a hasonló kezdeményezések számára.

[1] `http://linkeddata.org/`

DOI:10.15774/PPKE.ITK.2018.007

A létrehozott bővebb erőforrás segítségével az elméleti nyelvészek pontosabb képet kaphatnak az egyes igei szerkezetek működéséről, valamint az erőforrások felhasználhatók a jövőben szintaktikai és szemantikai elemzésre egyaránt.

IV. téziscsoport

A különböző nyelvi jelenségekből levont tanulságok nyomán bemutattam az Ana-Gramma elemző működését, az elmélet után a megvalósítás szempontjából is. Definiáltam a nyelvi jelenségek kezeléséhez használt ablakot, melynek ötlete a két fázisban működő *Sausage Machine*-ből (Frazier és J. D. Fodor 1978) származik. Az általam bemutatott ablak – mely a *Sausage Machine* első, PPP fázisának felel meg – és a rajta definiált keresőeljárások megoldást adnak a hatékony, emberi elemzőhöz hasonló balról jobbra elemzés számos problémájára.

9. Tézis. *Létrehoztam egy új megközelítésű (az ún. ablakra épülő) elemzési modell alapjait, amelynek elméletét társszerzővel közösen dolgoztam ki, és melynek segítségével a magyar nyelvű bemenet hatékonyan és az emberi feldolgozáshoz hasonlóan, szigorúan balról jobbra feldolgozható.*

A tézist alátámasztó közlemények: [14, 5, 21, 34, 25, 26]

A definiált ablakon működő eljárások közül ismertettem a jelöletlen (nominatívuszos) névszók egyértelműsítését és a jelölt (-nAk ragos) birtokos szerkezetek (Bánréti et al. 1992) kezelését. A módszerben a kétfázisú mondatelemzés első fázisában az előretekintő elemzési ablak segítségével megtörténik a testes esetrag nélküli elemek esetegyértelműsítése, melynek eredményeképpen tisztázódik a mondatbeli szerepük. Birtokos esetén, a bemutatott kereslet-kínálat keretrendszerben a birtok lesz az, amely birtokos-kereslettel él, amely kereslet kielégülésekor birtokos él jön létre a birtok és birtokosa között.

10. Tézis. *Létrehoztam egy az ablak segítségével a jelöletlen szerkezetek egyértelműsítését (például a magyar birtokos szerkezet és az alanyeset hatékony, valós idejű elkülönítését) elvégző algoritmust, melynek elméletét társszerzővel közösen közösen dolgoztam ki.*

A tézist alátámasztó közlemények: [5, 21, 25]

DOI:10.15774/PPKE.ITK.2018.007

Korpuszmérések alapján bizonyítottam, hogy az AnaGramma elemzőrendszer keretein belül a finit ige–igekötő kapcsolat létrehozása mellett (**Indig** és Vadász 2016b) az infinitívusz–igekötő és a finit ige–infinitívuszi vonzat kapcsolatok létrehozásához is elegendő a feltételezett két token méretű elemzési ablak használata. A tározó és az ablak segítségével a VFrame keresőeljárás a mondatban szereplő igei elemeket (finit és infinit igéket), valamint az igekötőket a megfelelő módon kapcsolja össze.

Az aktuális finit ige–igekötő–infinitívuszi vonzat kapcsolatok létrejötte után elindulnak a megfelelő vonzatkeresők, amelyek mind a tározóban, mind a mondat hátralévő részében keresik a vonzatkeret elemeit. Amennyiben a VFrame nem egyértelműsíti teljesen a vonzatkeretet (mert egy ige ugyanazzal a finit ige–igekötő–infinitívuszi vonzat viszonnyal többféle vonzatkerettel is rendelkezhet), akkor az összes ennek megfelelő vonzatkeret vonzatkeresője elindul. Ekkor a mondatban aktuálisan szereplő többi vonzat egyértelműsíti a vonzatkeretet.

11. Tézis. *Létrehoztam a VFrame eljárást, amelynek elméletét társszerzővel közösen dolgoztam ki, amivel a magyar nyelvben a helyes igekötő megtalálása az igekötők eloszlási mintájának ismeretében a lehetséges igei vonzatkeretek halmazának leszűkítésével történik.*

A tézist alátámasztó közlemények: [14, 26, 27]

Az eredményeim a magyar szövegek elemzésében változatosan alkalmazhatók. Például a bemutatott módszerekkel a mondatok mondatvázakká egyszerűsíthetők, amely megkönnyíti azok kezelését és a mondatok felépítésének és szerkezetének további kutatását. Az így kapott eredményekkel, már a teljes nyelvet leíró, az emberi elemzés működését figyelembe vevő elemzőrendszer hozható létre. Továbbá, azon pszicholingvisztikai kutatások számára, melyek az emberi elemzőt vizsgálják az általam definiált modell és az ennek nyomán elindult kutatások támpontot nyújthatnak és a két kutatás egymásnak adott visszajelezése által mindkét terület fejlődhet. Végül, a nyelvtechnológia alkalmazásaiban beleértve, de nem kizárólagosan a maximális főnévi csoportok pontosabb meghatározását – az esetegyértelműsítés által felszíni elemzéssel –, illetve az információ visszakeresést is javítani képes.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

7. fejezet

Az eredmények alkalmazási területei

„Csak még egy kérdés...”
(Colombo hadnagy)

A bemutatott eredmények frissességük miatt még nem kerültek széles körben alkalmazásra, viszont a főnévi csoportokkal és a szekvenciális címkézéssel kapcsolatos eredmények nagy érdeklődést vonzottak a nemzetközi konferenciákon. Úgy látom, hogy a jelenleg csak angol nyelvre megvizsgált eredmények némi változtatással átültethetőek magyar nyelvre, valamint más agglutináló nyelvekre is. Ezek egyike lehet az, hogy a meglévő szófaji egyértelműsítő módszerekkel egybeépítve a közvetlen összetevők és az NP-k határát jelölő annotációt is párhuzamosan el lehessen végezni. Az általam bemutatott enyhe lexikalizáció a dolgozatban bemutatott feladatokon túl számos feladatra általánosítható. A zárójelezés jólformáltságát ellenőrző metrika alkalmazása sok kellemetlenségtől kímélheti meg a jövő kutatóit minden szekvenciális címkézési feladat során.

Az összekapcsolt erőforrásokkal kapcsolatos elméleti eredményeim jól használhatóak később azok számára, akik hasonló erőforrás-összekapcsoláson gondolkoznak. Látható, hogy a rendszerben a szabályalapú összetevők túlsúlya miatt az ember által elkövethető hibák száma is nagyobb, ezért a tapasztalataimat érdemes figyelembe venni egy másik hasonló projekt előtt. A méréseimből az is látszik, hogy a jelenlegi szabályalapú és statisztikai erőforrások együttműködése egy jól használható rendszerként még nem megvalósított, így jobban járunk, ha csak a szabályalapú rendszereket használjuk.

DOI:10.15774/PPKE.ITK.2018.007

Az ismertetett munka alkalmazható például jó minőségű szemantikai információkat tartalmazó igei adatbázisok előállítására, melyek pontos szemantikai elemzést tesznek lehetővé és a jövőben számos elméleti nyelvészeti kutatás alapjául szolgálhatnak. Mindamellett már az igék igekötőinek keresésekor az általam kidolgozott elemzőmodellben fel is használtam, mely alkalmazás példaként szolgál az eredményeim alkalmazásához a számítógépes nyelvészet területén tevékenykedők számára. Távlati cél lehet az angol nyelvű erőforrásokból elérhető nyelvfüggetlen információ megbízható, automatikus átemelése magyar nyelvre a létrehozott ontológiák segítségével, de a neurális hálók előretörésével várhatóan a WordNet és a kézzel készített erőforrások háttérbe szorulnak a statisztikailag megalapozottabb erőforrásokkal szemben, így ezen szempontból a hosszútávú haszna kétséges.

A jelenlegi munkám részeként az eddigieknél nagyobb fedésű és kellően nagy pontosságú igei erőforrás (Manócska, lásd a 4.2.2. fejezet) előállításán és fejlesztésén munkálkodom. Az új erőforrásommal a fő célom, hogy egy korpuszból származó, statisztikailag jól alátámasztott, szintaktikai információt is tartalmazó adatbázist hozzak létre, mely teljesen reprodukálható és a későbbiekben bővíthető statisztikai alapon szerveződő szemantikai jellemzőkkel. Az általam készített ontológia széleskörű felhasználására, annak az ismertetett szabályalapú rendszerek miatti erős függése okán jelenleg nem látok esélyt, de specifikus esetben még jó szolgálatot tehet.

Az AnaGramma elemző architektúrájának tervezésekor felhasználtam a dolgozatban közölt többi eredményemet, melyek elméleti jelentősége nagyban hozzájárult a további új eredmények létrejöttéhez. Az utolsó fejezetben bemutatott eredményeim az elméleti nyelvészet szempontjából fontosak. Várható, hogy pszicholingvisztikai alkalmazásaik is lesznek, és további kutatások épülnek rájuk.

A főnévi csoportok pontos meghatározásával a mondatok mondatvázakká egyszerűsíthetők, mely várhatóan a kezelésüket és szerkezetük megértését nagyban könnyíteni fogja. A jövőben elkészülő elemzőrendszerrel széleskörű ipari alkalmazások is elképzelhetők. A nyelvtechnológia alkalmazói számára hasznos, hogy a maximális főnévi csoportok pontosabb meghatározása az esetegyértelműsítés által a felszíni elemzést, illetve az információ visszakeresést is javítani képes.

DOI:10.15774/PPKE.ITK.2018.007

Köszönetnyilvánítás

Megjelenési, de egyúttal fontossági sorrendben szeretném megköszönni először is szüleimnek, akik miatt eljutottam odáig, hogy legyen lehetőségem valamiről disszertációt írni. Sosem fogom tudni meghálálni társamnak és barátaimnak, azokat az erőfeszítéseket, amelyekkel az utolsó pillanatig tartották bennem a lelket, mivel ők a távoli jövőben sem terveznek doktori disszertációt írni. Nagyon köszönöm tanáraimnak, hogy a teljesítményem dacára felismerték az elszántságomat és előbb- utóbb átengedtek a vizsgákon. Köszönöm a konzulenseimnek, Hunyadvári Lászlónak, Garay Barnának és Prószéky Gábornak, hogy a nevüket adták a szárnypróbálgatásaimhoz. Köszönöm Roska Tamásnak, aki ahhoz az intézményhez adta a nevét, melyben annyi időt töltöttem, remélhetőleg nem hiába. Szeretném megköszönni a kollégáimnak a támogatást. Különös tekintettel Endrédy Istvánnak, aki olyan gellert adott a kutatásaimnak, amely a dolgozat első felét eredményezte. Köszönöm a lelkesedését és a gondos átolvasást Kalivoda Ágnesnek, aki amint egy nyelvtani hibát vétettem, javította legott. Köszönöm a szerzőtársaimnak, akik oly felsorolhatatlanul sokan vannak, hogy alig találtam bírálót a dolgozatomhoz. Köszönöm a bírálóimnak, hogy lelkiismeretesen végigolvasták a művem és minden apró hibára felhívták a figyelmem. A dolgozat jó minőségéért őket, a benne maradt hibákért teljes egészében engem lehet okolni. Köszönöm mindazoknak, akik olyan eredményeket publikáltak, amelyekben gondosan elrejtettek hibákat az utókornak, hogy azt felfedezve új erővel indulhassak a feladat megoldásának nyomára. Dolgozatomban szeretném én is ezt a gyakorlatot követni.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

A szerző közleményei

Nemzetközi folyóiratcikkek és könyvfejezetek

[1] Garay, Barnabás Miklós és **Balázs Indig** (2015.). „Chaos in Vallis' asymmetric Lorenz model for El Niño". *Chaos, Solitons & Fractals* 75.1., 253–262. old. ISSN: 0960-0779.

[2] **Indig**, **Balázs** (2017.a). „Less is More, More or Less... – Finding the Optimal Threshold for Lexicalization in Chunking". *Computación y Sistemas* 21.4.

[3] **Indig**, **Balázs** és István Endrédy (2018.). „Gut, Besser, Chunker – Selecting the best models for text chunking with voting". *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I (Lecture Notes in Artificial Intelligence)*. Szerk. Alexander Gelbukh. Cham: Springer International Publishing. Fej. 29, 409–423. old. ISBN: 978-3-319-75476-5. DOI: 10.1007/978-3-319-75477-2_29.

[4] **Indig**, **Balázs**, András Simonyi és Márton Miháltz (2018.). „Exploiting Linked Linguistic Resources for Semantic Role Labeling". *Human Language Technology. Challenges for Computer Science and Linguistics. 7th Language and Technology Conference, LTC 2015, Poznań, Poland, November 27-29, 2015. Revised Selected Papers (Lecture Notes in Artificial Intelligence 10930)*. Szerk. Zygmunt Vetulani, Joseph Mariani és Marek Kubis. Cham: Springer International Publishing. ISBN: 978-3-319-93781-6. DOI: 10.1007/978-3-319-93782-3.

[5] **Indig**, **Balázs**, Noémi Vadász és Ágnes Kalivoda (2016.). „Decreasing Entropy: How Wide to Open the Window?": *Theory and Practice of Natural Computing (Lecture Notes in Computer Science volume 10071)*. Szerk. Carlos Martín-Vide, Takaaki Mizuki és Miguel A. Vega-Rodríguez. Cham: Springer International Publishing, 137–148. old. ISBN: 978-3-319-49001-4. DOI: 0.1007/978-3-319-49001-4_11.

DOI:10.15774/PPKE.ITK.2018.007

Hazai folyóiratcikkek és könyvfejezetek

[6] Prószéky, Gábor és **Balázs Indig** (2015.a). „Magyar szövegek pszicholingvisztikai indíttatású elemzése számítógéppel". *Alkalmazott nyelvtudomány* 15.1-2., 29–44. old.

[7] Prószéky, Gábor, **Balázs Indig** és Noémi Vadász (2016.). „Performancia-alapú elemző magyar szövegek számítógépes megértéséhez". *''Szavad ne feledd!'': Tanulmányok Bánréti Zoltán tiszteletére*. Szerk. Bence Kas. Budapest: MTA Nyelvtudományi Intézet, 223–232. old.

Nemzetközi konferenciacikkek

[8] Endrédy, István és **Balázs Indig** (2015.). „HunTag3: a general-purpose, modular sequential tagger – chunking phrases in English and maximal NPs and NER for Hungarian". *7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. (Poznań, Poland, nov. 27.–2015). Poznań, Poland: Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, 213–218. old. ISBN: 978-83-932640-8-7.

[9] **Indig**, **Balázs** (2017.b). „Mosaic n-grams: Avoiding combinatorial explosion in corpus pattern mining for agglutinative languages". *8th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. (Poznań, Poland, nov. 17.–2017). Poznań, Poland: Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, 147–151. old. ISBN: 978-83-64864-94-0.

[10] **Indig**, **Balázs** (2018.b). „The stability of the parameter transformation with Zipfian distributions across corpora". *Computational Linguistics and Intelligent Text Processing: 19th International Conference, CICLing 2018, Hanoi, Vietnam, April 18–24, 2018, Revised Selected Papers, Part I (Lecture Notes in Artificial Intelligence)*. Szerk. Alexander Gelbukh. (Accepted, in press). Cham: Springer International Publishing.

[11] **Indig**, **Balázs**, Márton Miháltz és András Simonyi (2015.). „Exploiting Linked Linguistic Resources for Semantic Role Labeling". *7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. (Poznań, Poland, nov. 27.–2015). Poznań, Poland: Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, 140–144. old. ISBN: 978-83-932640-8-7.

[12] **Indig**, **Balázs**, Márton Miháltz és András Simonyi (2016.). „Mapping Ontologies Using Ontologies: Cross-lingual Semantic Role Information Transfer". *Proceedings of the Tenth International Conference on Language Re-*

DOI:10.15774/PPKE.ITK.2018.007

sources and Evaluation (LREC 2016). Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk és Stelios Piperidis. Portorož, Slovenia: European Language Resources Association (ELRA), 2425–2430. old. ISBN: 978-2-9517408-9-1.

[13] **Indig**, **Balázs**, András Simonyi és Noémi Ligeti-Nagy (2018.). „What's Wrong, Python? – A Visual Differ and Graph Library for NLP in Python". *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Szerk. Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis és Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.

[14] **Indig**, **Balázs** és Noémi Vadász (2016.b). „Windows in Human Parsing – How Far can a Preverb Go?": *Proceedings of the Tenth International Conference on Natural Language Processing (HrTAL2016) 2016, Dubrovnik, Croatia, September 29-October 1., 2016*. Szerk. Marko Tadić és Božo Bekavac. (Accepted, in press).

[15] Miháltz, Márton, Bálint Sass és **Balázs Indig** (2013.). „What Do We Drink? Automatically Extending Hungarian WordNet With Selectional Preference Relations". *Proceedings of the Joint Symposium on Semantic Processing: Textual Inference and Structures in Corpora*. (nov. 20.–2013). Szerk. Octavian Popescu és Alberto Lavelli. Trento, Italy: Association for Computational Linguistics (ACL), 105–109. old. ISBN: 978-1-6299353-9-3.

[16] Váradi, Tamás, Eszter Simon, Bálint Sass, Iván Mittelholcz, Attila Novák, **Balázs Indig**, Richárd Farkas és Veronika Vincze (2018.). „E-magyar – A Digital Language Processing System". *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Szerk. Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis és Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA). ISBN: 979-10-95546-00-9.

Hazai konferenciacikkek

[17] **Indig**, **Balázs** (2013.b). „PureToken: egy új tokenizáló eszköz". *IX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Inté-

DOI:10.15774/PPKE.ITK.2018.007

zet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 305–309. old.

[18] **Indig**, **Balázs** (2018.a). „Közös crawlnak is egy korpusz a vége – Korpuszépítés a CommonCrawl .hu domainjából". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 125–135. old.

[19] **Indig**, **Balázs**, László János Laki és Gábor Prószéky (2016.). „Mozaik nyelvmodell az AnaGramma elemzőhöz". *XII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2016)*. Szerk. Attila Tanács, Viktor Varga és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 260–270. old.

[20] **Indig**, **Balázs** és Gábor Prószéky (2013.). „Ismeretlen szavak helyes kezelése kötegelt helyesírás-ellenőrző programmal". *IX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 310–317. old.

[21] Ligeti-Nagy, Noémi, Noémi Vadász, Andrea Dömötör és **Balázs Indig** (2018.). „Nulla vagy semmi? Esetegyértelműsítés az ablakban". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 25–37. old.

[22] Miháltz, Márton, **Balázs Indig** és Gábor Prószéky (2015.). „Igei vonzatkeretek és tematikus szerepek felismerése nyelvi erőforrások összekapcsolásával egy kereslet-kínálat elvű mondatelemzőben". *XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015)*. Szerk. Attila Tanács, Viktor Varga és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 298–302. old.

[23] Novák, Attila, György Orosz és **Balázs Indig** (2011.). „Javában taggelünk". *VIII.. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 310–317. old.

[24] Prószéky, Gábor, **Balázs Indig**, Márton Miháltz és Bálint Sass (2014.). „Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé". *X. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2014)*. Szerk. Attila Tanács, Viktor Varga és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 79–87. old.

DOI:10.15774/PPKE.ITK.2018.007

[25] Vadász, Noémi és **Balázs Indig** (2018.). „A birtokos esete az ablakkal". *LingDok: nyelvész-doktoranduszok dolgozatai*. Szerk. György Scheibl. Szegedi Tudományegyetem. Nyelvtudományi Doktori Iskola, old. 85–99.

[26] Vadász, Noémi, Ágnes Kalivoda és **Balázs Indig** (2017.). „Ablak által világosan – Vonzatkeret-egyértelműsítés az igekötők és az infinitívuszi vonzatok segítségével". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 3–12. old.

[27] Vadász, Noémi, Ágnes Kalivoda és **Balázs Indig** (2018.). „Egy egységesített magyar igei vonzatkerettár építése és felhasználása". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 3–15. old.

[28] Váradi, Tamás, Eszter Simon, Bálint Sass, Mátyás Gerőcs, Iván Mittelholcz, Attila Novák, **Balázs Indig**, Gábor Prószéky, Richárd Farkas és Veronika Vincze (2017.). „Az `e-magyar` digitális nyelvfeldolgozó rendszer". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 49–60. old.

Egyéb közlemények

[29] **Indig**, **Balázs** (2013.a). „An extended spell checker for unknown words". *Pázmány Péter Catholic University PhD Proceedings* 8., 29–32. old.

[30] **Indig**, **Balázs** (2014.a). „Towards a Psycholinguistically Motivated Performance-Based Parsing Model". *PhD Proceedings Annual Issues of the Doctoral School Faculty of Information Technology and Bionics* 2014., 133–136. old.

[31] **Indig**, **Balázs** (2014.b). „Towards recognizing thematic roles for verbal frames by linking two independent language resources for a parser based on the supply and demand paradigm". *PhD Proceedings Annual Issues of the Doctoral School Faculty of Information Technology and Bionics* 2015., 159–161. old.

[32] **Indig**, **Balázs** és Noémi Vadász (2016.a). *POS Comes with Parsing: a Refined Word Categorisation Method*. Konfereciaabsztrakt (konferenciakötetbe nem került), 4th International Conference on Statistical Language and Speech Processing (SLSP 2016), Csehország, Plzeň, 2016. október 11-12.

DOI:10.15774/PPKE.ITK.2018.007

Pilseň, Czech Republic. URL: http://grammars.grlmc.com/SLSP2016/Download/slides/pos-comes-with-parsing-abstract.pdf.

[33] **Indig**, **Balázs**, Noémi Vadász és Ágnes Kalivoda (2017.). *Manócska – integrált igeivonzatkeret-adatbázis*. URL: https://github.com/ppke-nlpg/manocska.

[34] Prószéky, Gábor és **Balázs Indig** (2015.b). *Natural parsing: a psycholinguistically motivated computational language processing model*. Konferecia-absztrakt (konferenciakötetbe nem került), 4th International Conference on the Theory and Practice of Natural Computing (TPNC 2015), Spanyolország, Astruias, Mieres, 2015. december 15-16. Mieres, Astruias, Spain. URL: http://grammars.grlmc.com/TPNC2015/Slides/d1s503natural_parsing_abstract.pdf.

DOI:10.15774/PPKE.ITK.2018.007

Hivatkozások

Csendes, Dóra, János Csirik, Tibor Gyimóthy és András Kocsor (2005.). „The Szeged Treebank". *Text, Speech and Dialogue: 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, September 12-15, 2005. Proceedings*. Szerk. Václav Matoušek, Pavel Mautner és Tomáš Pavelka. Berlin, Heidelberg: Springer Berlin Heidelberg, 123–131. old. ISBN: 978-3-540-31817-0.

Csendes, Dóra, Csaba Hatvani, Zoltán Alexin, János Csirik, Tibor Gyimóthy, Gábor Prószéky és Tamás Váradi (2003.). „Kézzel annotált magyar nyelvi korpusz: a Szeged Korpusz". *I. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)*. Szerk. Zoltán Alexin és Dóra Csendes. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 238–245. old.

Gyalus, Márk (2018.). „Magyar főnévi csoportok gépi azonosítása a gyakorlatban (Nyelvészeti problémák az automatikus csonkolóprogramokkal)". Mesterszakos szakdolgozat. PPKE-BTK.

Bánréti, Zoltán, István Kenesei, András Komlósy, Tibor Laczkó és Anna Szabolcsi (1992.). *Strukturális magyar nyelvtan I: Mondattan*. Szerk. Ferenc Kiefer és Zsófia Róbert. Akadémiai Kiadó. ISBN: 963-05-6468-8.

Berners-Lee, Tim, James Hendler és Ora Lassila (2001.). „The Semantic Web". *Scientific American* 284.5., 34–43. old.

Bies, Ann, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz és Britta Schasberger (1995.). „Bracketing guidelines for Treebank II style Penn Treebank project". *University of Pennsylvania* 97., 100. old.

Björkelund, Anders, Love Hafdell és Pierre Nugues (2009.). „Multilingual semantic role labeling". *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, 43–48. old.

Bohnet, Bernd (2010.). „Very high accuracy and fast dependency parsing is not a contradiction". *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 89–97. old.

DOI:10.15774/PPKE.ITK.2018.007

Brants, Thorsten (2000.). „T'n'T: a statistical part-of-speech tagger". *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 224–231. old.

Brants, Thorsten és Matthew Crocker (2000.). „Probabilistic parsing and psychological plausibility". *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Saarbrücken:Association for Computational Linguistics, 111–117. old.

Zsibrita, János, Veronika Vincze és Richárd Farkas (2013.). „MAGYARLANC: A Toolkit for Morphological and Dependency Parsing of Hungarian". *Proceedings of RANLP 2013. Hissar, Bulgária, 2013.09.08-2013.09.13.* Old. 763–771.

Chomsky, Noam (1957.). *Syntactic structures*. The Hague:Mouton.

Costa-jussà, Marta R., Carlos Escolano és José A. R. Fonollosa (2017.). „Byte-based Neural Machine Translation". *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. Copenhagen, Denmark: Association for Computational Linguistics, 154–158. old. URL: `http://aclweb.org/anthology/W17-4123`.

De Marneffe, Marie-Catherine, Bill MacCartney és Christopher D Manning (2006.). „Generating Typed Dependency Parses from Phrase Structure Parses". *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*. 6. Genoa, Italy: European Language Resources Association (ELRA), 449–454. old.

Domaradzki, Mikołaj (2007.). „Cognitive Critique of Generative Grammar". *Lingua ac Communitas* 17., 39–58. old.

É. Kiss, Katalin (2007.). „Az ige utáni szabad szórend magyarázata". *Nyelvtudományi Közlemények* 104., 124–152. old.

Endrédy, István (2014.). „Corpus driven research: ideas and attempts". *PhD Proceedings Annual Issues of the Doctoral School Faculty of Information Technology and Bionics* 2014., 137–140. old.

Endrédy, István (2016.). „Nyelvtechnológiai algoritmusok korpuszok automatikus építéséhez és pontosabb feldolgozásukhoz". PhD dissz. Budapest: PPKE-ITK.

Endrédy, István és Balázs Indig (2015.). „HunTag3: a general-purpose, modular sequential tagger – chunking phrases in English and maximal NPs and NER for Hungarian". *7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. (Poznań, Poland, nov. 27.–2015). Poznań, Poland: Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, 213–218. old. ISBN: 978-83-932640-8-7.

Erjavec, Tomaž (2010.). „MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora". *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*. Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner és Daniel

DOI:10.15774/PPKE.ITK.2018.007

Tapias. Valletta, Malta: European Language Resources Association (ELRA). ISBN: 2-9517408-6-7.

Farajian, M. Amin, Marco Turchi, Matteo Negri és Marcello Federico (2017.). „Multi-Domain Neural Machine Translation through Unsupervised Adaptation". *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, 127–137. old. URL: http://aclweb.org/anthology/W17-4713.

Fodor, Jerry és Ernie Lepore (2004.). „Out of Context". *Proceedings and Addresses of the American Philosophical Association* 78.2., 77–94. old. ISSN: 0065972X.

Forney, G David (1973.). „The viterbi algorithm". *Proceedings of the IEEE* 61.3., 268–278. old.

Frazier, Lyn és Janet Dean Fodor (1978.). „The Sausage Machine: A New Two-Stage Parsing Model". *Cognition* 6.4., 291–325. old.

Gábor, Kata, Enikő Héja, Judit Kuti, Viktor Nagy és Tamás Váradi (2008.). „A lexikon a nyelvtechnológiában". *Strukturális magyar nyelvtan IV: A lexikon szerkezete*. Szerk. Ferenc Kiefer. 4. Akadémiai Kiadó, Budapest, 853–893. old.

Grice, H Paul és G. Harman (1975.). *Logic and conversation*. University of California, Berkeley, old. 64–75.

Haarslev, Volker, Kay Hidde, Ralf Möller és Michael Wessel (2012.). „The RacerPro knowledge representation and reasoning system". *Semantic Web Journal* 3.3., 267–277. old.

Halácsy, Péter, András Kornai, Németh László, Rung András, István Szakadát és Trón Viktor (2004.). „Creating open language resources for Hungarian". *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*. Szerk. Calzolari N, old. 203–210.

Halácsy, Péter, András Kornai és Csaba Oravecz (2007.). „HunPOS: An Open Source Trigram Tagger". *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. ACL '07. Prague, Czech Republic: Association for Computational Linguistics, 209–212. old.

Handler, Abram (2014.). „An empirical study of semantic similarity in WordNet and Word2Vec". Mesterszakos szakdolgozat. University of New Orleans.

Hobbs, Jerry R. (1985.). „Ontological Promiscuity". *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics*. ACL '85. Chicago, Illinois: Association for Computational Linguistics, 60–69. old. DOI: 10.3115/981210.981218. URL: https://doi.org/10.3115/981210.981218.

Indig, Balázs (2013.). „PureToken: egy új tokenizáló eszköz". *X. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2013)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 305–309. old.

Indig, Balázs (2017.). „Less is More, More or Less... – Finding the Optimal Threshold for Lexicalization in Chunking". *Computación y Sistemas* 21.4.

DOI:10.15774/PPKE.ITK.2018.007

Indig, Balázs (2018.). „Közös crawlnak is egy korpusz a vége – Korpuszépítés a CommonCrawl .hu domainjából". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 125–135. old.

Indig, Balázs és István Endrédy (2018.). „Gut, Besser, Chunker – Selecting the best models for text chunking with voting". *Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I (Lecture Notes in Artificial Intelligence)*. Szerk. Alexander Gelbukh. Cham: Springer International Publishing. Fej. 29, 409–423. old. ISBN: 978-3-319-75476-5. DOI: 10.1007/978-3-319-75477-2_29.

Indig, Balázs, Márton Miháltz és András Simonyi (2016.). „Mapping Ontologies Using Ontologies: Cross-lingual Semantic Role Information Transfer". *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk és Stelios Piperidis. Portorož, Slovenia: European Language Resources Association (ELRA), 2425–2430. old. ISBN: 978-2-9517408-9-1.

Indig, Balázs, András Simonyi és Márton Miháltz (2018.). „Exploiting Linked Linguistic Resources for Semantic Role Labeling". *Human Language Technology. Challenges for Computer Science and Linguistics. 7th Language and Technology Conference, LTC 2015, Poznań, Poland, November 27-29, 2015. Revised Selected Papers (Lecture Notes in Artificial Intelligence 10930)*. Szerk. Zygmunt Vetulani, Joseph Mariani és Marek Kubis. Cham: Springer International Publishing. ISBN: 978-3-319-93781-6. DOI: 10.1007/978-3-319-93782-3.

Indig, Balázs és Noémi Vadász (2016.a). *POS Comes with Parsing: a Refined Word Categorisation Method*. Konfereciaabsztrakt (konferenciakötetbe nem került), 4th International Conference on Statistical Language and Speech Processing (SLSP 2016), Csehország, Plzeň, 2016. október 11-12. Pilseň, Czech Republic. URL: http://grammars.grlmc.com/SLSP2016/Download/slides/pos-comes-with-parsing-abstract.pdf.

Indig, Balázs és Noémi Vadász (2016.b). „Windows in Human Parsing – How Far can a Preverb Go?": *Proceedings of the Tenth International Conference on Natural Language Processing (HrTAL2016) 2016, Dubrovnik, Croatia, September 29-October 1., 2016*. Szerk. Marko Tadić és Božo Bekavac. (Accepted, in press).

Indig, Balázs, Noémi Vadász és Ágnes Kalivoda (2016.). „Decreasing Entropy: How Wide to Open the Window?": *Theory and Practice of Natural Computing (Lecture Notes in Computer Science volume 10071)*. Szerk. Carlos Martín-

DOI:10.15774/PPKE.ITK.2018.007

Vide, Takaaki Mizuki és Miguel A. Vega-Rodríguez. Cham: Springer International Publishing, 137–148. old. ISBN: 978-3-319-49001-4. DOI: 0.1007/978-3-319-49001-4_11.

Indig, Balázs, Noémi Vadász és Ágnes Kalivoda (2017.). *Manócska – integrált igeivonzatkeret-adatbázis*. URL: https://github.com/ppke-nlpg/manocska.

Kalivoda, Ágnes (2016.). „A magyar igei komplexumok vizsgálata". Mesterszakos szakdolgozat. PPKE-BTK. URL: https://github.com/kagnes/hungarian_verbal_complex.

Kalivoda, Ágnes (2017.). „Hungarian particle verbs in a corpus-driven approach". *Computational Linguistics and Intelligent Text Processing - 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Proceedings*. (Accepted, in press).

Kálmán C., György, László Kálmán, Ádám Nádasdy és Gábor Prószéky (1989.). „A magyar segédigék rendszere". *Általános nyelvészeti tanulmányok – Tanulmányok a magyar mondattan köréből* 17.1. Szerk. Zsigmond Telegdi és Ferenc Kiefer, 49–103. old.

Kornai, András (1985.). „The internal structure of Noun Phrases". *Approaches to Hungarian* 1., 79–92. old.

Kornai, András, Dávid Márk Nemeskey és Gábor Recski (2016.). „Detecting Optional Arguments of Verbs". *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk és Stelios Piperidis. Portorož, Slovenia: European Language Resources Association (ELRA). ISBN: 978-2-9517408-9-1.

Kornai, András, Péter Rebrus, Péter Vajda, Péter Halácsy, András Rung és Viktor Trón (2004.). „Általános célú morfológiai elemző kimeneti formalizmusa". *II. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2004)*. Szerk. Zoltán Alexin és Dóra Csendes. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 172–176. old.

Lafferty, John D., Andrew McCallum és Fernando C. N. Pereira (2001.). „Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 282–289. old. ISBN: 1-55860-778-1.

Laki, László János, György Orosz és Attila Novák (2013.). „HuLaPos 2.0 – Decoding Morphology". *Proceedings of the Advances in Artificial Intelligence and Its Applications: 12th Mexican International Conference on Artificial Intelligence (MICAI 2013) Part I, Mexico City, Mexico, November 24-30, 2013*. Szerk. Félix Castro, Alexander Gelbukh és Miguel González, 294–305. old. ISBN: 978-3-642-45114-0.

DOI:10.15774/PPKE.ITK.2018.007

Liebig, Thorsten, Marko Luther, Olaf Noppens és Michael Wessel (2011.). „OWLlink". *Semantic Web – Interoperability, Usability, Applicability* 2.1., 23–32. old.

Ligeti-Nagy, Noémi (2016.). „A főnévi csoportok és ami utánuk marad. Automatikus szintagmakinyerés magyar szövegekből". *Távlatok a mai magyar alkalmazott nyelvészetben*. Tinta Könyvkiadó, 249–260. old.

Ligeti-Nagy, Noémi, Noémi Vadász, Andrea Dömötör és Balázs Indig (2018.). „Nulla vagy semmi? Esetegyértelműsítés az ablakban". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 25–37. old.

Lindén, Krister, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A. Pirinen és Miikka Silfverberg (2013.). „HFST — A System for Creating NLP Tools". *Systems and Frameworks for Computational Morphology*. Szerk. Cerstin Mahlow és Michael Piotrowski. Berlin, Heidelberg: Springer Berlin Heidelberg, 53–71. old. ISBN: 978-3-642-40486-3.

Loper, Edward és Steven Bird (2002.). „NLTK: The Natural Language Toolkit". *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Loper, Edward, Szu-Ting Yi és Martha Palmer (2007.). „Combining lexical resources: mapping between PropBank and VerbNet". *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg*, 118–128. old.

Manning, Christopher D, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard és David McClosky (2014.). „The Stanford CoreNLP natural language processing toolkit." *Proceedings of ACL 2014: System Demonstrations*, 55–60. old.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard és David McClosky (2014.). „The Stanford CoreNLP Natural Language Processing Toolkit". *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. old.

McCallum, Andrew, Dayne Freitag és Fernando C. N. Pereira (2000.). „Maximum Entropy Markov Models for Information Extraction and Segmentation". *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 591–598. old. ISBN: 1-55860-707-2.

McConkie, George W. és Keith Rayner (1976.). „Asymmetry of the perceptual span in reading". *Bulletin of the Psychonomic Society* 8.5., 365–368. old. ISSN: 0090-5054. DOI: 10.3758/BF03335168. URL: https://doi.org/10.3758/BF03335168.

Miháczi, András, László Németh és Miklós Rácz (2003.). „Magyar szövegek természetes nyelvi előfeldolgozása". *I. Magyar Számítógépes Nyelvészeti Konferencia*

DOI:10.15774/PPKE.ITK.2018.007

(MSZNY 2003). Szerk. Zoltán Alexin és Dóra Csendes. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 38–43. old.

Miháltz, Márton (2011.). „Magyar NP-felismerők összehasonlítása". *VIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 333–335. old.

Miháltz, Márton, Csaba Hatvani, Judit Kuti, György Szarvas, János Csirik, Gábor Prószéky és Tamás Váradi (2008.). „Methods and Results of the Hungarian WordNet Project". *Proceedings of The Fourth Global WordNet Conference*. Szeged, Hungary, 311–321. old.

Miller, George A. (1995.). „WordNet: A Lexical Database for English". *Commun. ACM* 38.11., 39–41. old. ISSN: 0001-0782.

Mittelholcz, Iván (2017.). „`emToken`: Unicode-képes tokenizáló magyar nyelvre". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 61–69. old.

Mogren, Olof és Richard Johansson (2017.). „Character-based recurrent neural networks for morphological relational reasoning". *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. Copenhagen, Denmark: Association for Computational Linguistics, 57–63. old. URL: `http://www.aclweb.org/anthology/W17-4108`.

Molina, Antonio és Ferran Pla (2002.). „Shallow parsing using specialized HMMs". *Journal of Machine Learning Research* 2., 595–613. old.

Nivre, Joakim (2006.). *Inductive dependency parsing*. Springer.

Novák, Attila (2003.). „Milyen a jó humor?": *I. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)*. Szerk. Zoltán Alexin és Dóra Csendes. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 138–145. old.

Novák, Attila, György Orosz és Balázs Indig (2011.). „Javában taggelünk". *VIII.. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 310–317. old.

Novák, Attila, Péter Rebrus és Zsófia Ludányi (2017.). „Az `emMorph` morfológiai elemző annotációs formalizmusa". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 70–78. old.

Oflazer, Kemal (2003.). „Dependency parsing with an extended finite-state approach". *Computational Linguistics* 29.4., 515–544. old.

DOI:10.15774/PPKE.ITK.2018.007

Okazaki, Naoaki (2007.). *CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*. URL: `http://www.chokkan.org/software/crfsuite/`.

Oravecz, Csaba, Tamás Váradi és Bálint Sass (2014.). „The Hungarian Gigaword Corpus". *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*. Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk és Stelios Piperidis. Reykjavik, Iceland: European Language Resources Association (ELRA). ISBN: 978-2-9517408-8-4.

Orosz, György és Attila Novák (2013.). „PurePos 2.0: a hybrid tool for morphological disambiguation". *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, 539–545. old.

Pedersen, Ted, Siddharth Patwardhan és Jason Michelizzi (2004.). „WordNet:: Similarity: measuring the relatedness of concepts". *Demonstration papers at HLT-NAACL 2004*. Association for Computational Linguistics, 38–41. old.

Pléh, Csaba (1999.). *Mondatmegértés a magyar nyelvben*. Osiris Kiadó, Budapest.

Pléh, Csaba (2014.). *Pszicholingvisztika: magyar pszicholingvisztikai kézikönyv*. Akadémiai Kiadó.

Prószéky, Gábor (2000.). „Számítógépes morfológia". *Strukturális magyar nyelvtan III: Morfológia*. Szerk. Ferenc Kiefer és Zoltán Bánréti. 3. Akadémiai Kiadó, Budapest, 1021–1064. old.

Prószéky, Gábor és Balázs Indig (2015.a). „Magyar szövegek pszicholingvisztikai indíttatású elemzése számítógéppel". *Alkalmazott nyelvtudomány* 15.1-2., 29–44. old.

Prószéky, Gábor és Balázs Indig (2015.b). *Natural parsing: a psycholinguistically motivated computational language processing model*. Konfereciaabsztrakt (konferenciakötetbe nem került), 4th International Conference on the Theory and Practice of Natural Computing (TPNC 2015), Spanyolország, Astruias, Mieres, 2015. december 15-16. Mieres, Astruias, Spain. URL: `http://grammars.grlmc.com/TPNC2015/Slides/d1s503natural_parsing_abstract.pdf`.

Prószéky, Gábor, Balázs Indig, Márton Miháltz és Bálint Sass (2014.). „Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé". *X. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2014)*. Szerk. Attila Tanács, Viktor Varga és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 79–87. old.

Prószéky, Gábor, Balázs Indig és Noémi Vadász (2016.). „Performanciaalapú elemző magyar szövegek számítógépes megértéséhez". *''Szavad ne feledd!'': Tanulmányok Bánréti Zoltán tiszteletére*. Szerk. Bence Kas. Budapest: MTA Nyelvtudományi Intézet, 223–232. old.

DOI:10.15774/PPKE.ITK.2018.007

Prószéky, Gábor, Ilona Koutny és Balázs Wacha (1989.). „A dependency syntax of Hungarian". *Metataxis in Practice (Dependency Syntax for Multilingual Machine Translation)*. Szerk. D. Maxwell és K. Schubert, 151–181. old.

Prószéky, Gábor, Márton Miháltz és Judit Kuti (2013.). „Lexikális szemantika: a számítógépes nyelvészet és a pszicholingvisztika határán". *Általános Nyelvészeti Tanulmányok* XXV., 143–172. old.

Prószéky, Gábor és László Tihanyi (2002.). „MetaMorpho: A pattern-based machine translation system". *Proceedings of the 24th Translating and the Computer Conference*, 19–24. old.

Prószéky, Gábor, László Tihanyi és Gábor Ugray (2004.). „Moose: A robust high-performance parser and generator". *Proceedings of the 9th Workshop of the European Association for Machine Translation*. (La Valletta, Malta), 138–142. old.

Ramshaw, Lance A. és Mitchell P. Marcus (1995.). „Text Chunking Using Transformation-Based Learning". *Proceedings of the Third ACL Workshop on Very Large Corpora*. Cambridge, MA, USA, 82–94. old.

Ratnaparkhi, Adwait (1996.). „A Maximum Entropy Model for Part-Of-Speech Tagging". *Conference on Empirical Methods in Natural Language Processing*, 133–142. old.

Rayner, Keith (1998.). „Eye movements in reading and information processing: 20 years of research." *Psychological bulletin* 124.3., 372. old.

Recski, Gábor (2011.). „A sekély mondattani elemzés további lépései". *VIII.. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2011)*. Szerk. Attila Tanács és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 310–317. old.

Recski, Gábor (2014.). „Hungarian Noun Phrase Extraction Using Rule-based and Hybrid Methods". *Acta Cybernetica* 21.3.

Recski, Gábor és Dániel Varga (2009.). „A Hungarian NP Chunker". *The Odd Yearbook. ELTE SEAS Undergraduate Papers in Linguistics*, 87–93. old.

Recski, Gábor és Dániel Varga (2012.). „Magyar főnévi csoportok azonosítása". *Általános Nyelvészeti Tanulmányok* XXIV. Szerk. Gábor Prószéky, Tamás Váradi és István Kenesei.

Roth, Michael és Mirella Lapata (2016.). „Neural Semantic Role Labeling with Dependency Path Embeddings". *Proceedings of ACL 2016, Berlin*, 1192–1202. old.

Roth, Michael és Mirella Lapata (2017.). *PathLSTM*. `https://github.com/microth/PathLSTM`. GitHub repository.

Sass, Bálint (2009.). „A Unified Method for Extracting Simple and Multiword Verbs with Valence Information and Application for Hungarian." *Recent Advances in Natural Language Processing RANLP*. Szerk. Galia Angelova, Ka-

DOI:10.15774/PPKE.ITK.2018.007

lina Bontcheva, Ruslan Mitkov, Nicolas Nicolov és Nikolai Nikolov. Borovets, Bulgaria: RANLP, 399–403. old.

Sass, Bálint (2011.). „Igei szerkezetek gyakorisági szótára – Egy automatikus lexikai kinyerő eljárás és alkalmazása". PhD dissz. Pázmány Péter Katolikus Egyetem ITK.

Sass, Bálint (2015.). „28 millió szintaktikailag elemzett mondat és 500000 igei szerkezet". *XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015)*. Szerk. Attila Tanács, Viktor Varga és Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 399–403. old.

Sass, Bálint, Tamás Váradi, Júlia Pajzs és Margit Kiss (2010.). *Magyar igei szerkezetek – A leggyakoribb vonzatok és szókapcsolatok szótára.* Budapest: Tinta Könyvkiadó.

Schuler, Karin Kipper (2005.). „VerbNet: A broad-coverage, comprehensive verb lexicon". PhD dissz.

Shen, Hong és Anoop Sarkar (2005.). „Voting Between Multiple Data Representations for Text Chunking". *Proceedings of the Advances in Artificial Intelligence, 18th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2005, Victoria, Canada, May 9-11, 2005*. Szerk. Balázs Kégl és Guy Lapalme. 3501. Lecture Notes in Computer Science. Springer, 389–400. old.

Simon, Eszter (2013.). „Approaches to hungarian named entity recognition". PhD dissz. Budapesti Műszaki és Gazdaságtudományi Egyetem.

Tjong Kim Sang, Erik F. és Sabine Buchholz (2000.). „Introduction to the CoNLL-2000 Shared Task: Chunking". *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*. ConLL '00. Lisbon, Portugal: Association for Computational Linguistics, 127–132. old.

Trón, Viktor, András Kornai, György Gyepesi, László Németh, Péter Halácsy és Dániel Varga (2005.). „Hunmorph: open source word analysis". *Proceedings of the Workshop on Software*. Association for Computational Linguistics, 77–85. old.

Tufis, Dan, Dan Cristea és Sofia Stamou (2004.). „BalkaNet: Aims, methods, results and perspectives. a general overview". *Romanian Journal of Information science and technology* 7.1-2., 9–43. old.

Vadász, Noémi és Balázs Indig (2018.). „A birtokos esete az ablakkal". *LingDok: nyelvész-doktoranduszok dolgozatai*. Szerk. György Scheibl. Szegedi Tudományegyetem. Nyelvtudományi Doktori Iskola, old. 85–99.

Vadász, Noémi, Ágnes Kalivoda és Balázs Indig (2017.). „Ablak által világosan – Vonzatkeret-egyértelműsítés az igekötők és az infinitívuszi vonzatok segítségével". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*.

DOI:10.15774/PPKE.ITK.2018.007

Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 3–12. old.

Vadász, Noémi, Ágnes Kalivoda és Balázs Indig (2018.). „Egy egységesített magyar igei vonzatkerettár építése és felhasználása". *XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem, Informatikai Tanszékcsoport, 3–15. old.

Váradi, Tamás (2002.). „The Hungarian National Corpus". *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002) European Language Resources Association, Paris*, 385–389. old.

Váradi, Tamás (2003.). „Shallow parsing of Hungarian business news". *Proceedings of the Corpus Linguistics 2003 Conference*, 845–851. old.

Váradi, Tamás, Eszter Simon, Bálint Sass, Mátyás Gerőcs, Iván Mittelholcz, Attila Novák, Balázs Indig, Gábor Prószéky, Richárd Farkas és Veronika Vincze (2017.). „Az `e-magyar` digitális nyelvfeldolgozó rendszer". *XIII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2017)*. Szerk. Veronika Vincze. Szegedi Tudományegyetem Informatikai Intézet. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 49–60. old.

Varga, Dániel és Eszter Simon (2007.). „Hungarian Named Entity Recognition with a Maximum Entropy Approach". *Acta Cybernetica* 18.2., 293–301. old. ISSN: 0324-721X.

Vincze, Veronika, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin és János Csirik (2010.). „Hungarian Dependency Treebank". *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*. Szerk. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner és Daniel Tapias. Valletta, Malta: European Language Resources Association (ELRA), 1855–1862. old. ISBN: 2-9517408-6-7.

Visser, Eelco (1997.). *Scannerless generalized-LR parsing*. Tech. rep. Universiteit van Amsterdam. Programming Research Group.

Vossen, Piek, Laura Bloksma, Horacio Rodriguez, Salvador Climent, Nicoletta Calzolari, Adriana Roventini, Francesca Bertagna, Antonietta Alonge és Wim Peters (1998.). *The EuroWordNet base Concepts and Top Ontology*. Tech. rep.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

Függelék

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

A. függelék

Közvetlen összetevők keresése angol nyelven

A méréseim eredményei a lexikalizáció definiált szintjein a különböző címkéző programokkal az öt főbb `IOB` reprezentáción magukban és a reprezentációk közötti szavazással.

	IOB1	**IOB2**	**IOE1**	**IOE2**	**IOBES**
T'n'T	`[82,23]`	`[84,27]`	`[78,83]`	`[81,45]`	`[86,95]`
NLTK T'n'T	82,10	84,84	78,75	87,58	87,58
PurePOS	84,28	84,92	84,39	86,75	87,85
HunTag3 Bigram	91,73	92,40	91,85	92,37	93,26
HunTag3 Trigram	92,00	92,75	92,05	92,81	93,47
HunTag3 CRFSuite	92,41	92,84	92,10	92,77	93,41
Hivatalos CRFSuite	*92,84*	*93,40*	*92,92*	*93,25*	*93,79*

A.1. táblázat. Megvizsgáltam minden címkézőt magában, **lexikalizáció nélkül**. Az SS05 módszer reprodukált eredményei `[szögletes zárójelbe rakott írógép stílusban]` láthatóak, a legjobb F-mértékek (%) pedig *dőlt betűvel* szedettek. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

DOI:10.15774/PPKE.ITK.2018.007

	IOB1	**IOB2**	**IOE1**	**IOE2**	**IOBES**
T'n'T	87,39	88,67	87,06	88,95	90,23
NLTK T'n'T	87,33	88,69	87,00	89,14	90,66
PurePOS	88,33	88,82	88,50	90,27	91,04
HunTag3 Bigram					
HunTag3 Trigram					
HunTag3 CRFSuite	93,20	93,85	93,35	**94,13**	**94,28**
Hivatalos CRFSuite	***94,13***	***94,70***	***94,09***	***94,61***	***94,94***

A.2. táblázat. Megvizsgáltam minden címkézőt magában, **enyhe lexikalizációval**. A legjobb F-mértékek (%) *dőlt betűvel* szedettek és minden 94% fölötti F-mérték **félkövérrel** szedett. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

	IOB1	**IOB2**	**IOE1**	**IOE2**	**IOBES**
T'n'T	`[91,12]`	`[91,33]`	`[91,17]`	`[91,36]`	`[91,43]`
NLTK T'n'T	91,00	91,32	91,04	91,40	91.61
PurePOS	91,42	91,34	91,35	91,58	91,65
HunTag3 Bigram					
HunTag3 Trigram					
HunTag3 CRFSuite	92,64	93,21	92,94	93,44	93,35
Hivatalos CRFSuite	**93,65**	**94,03**	**93,65**	**94,12**	**94,16**

A.3. táblázat. Megvizsgáltam minden címkézőt magában, **teljes lexikalizációval**. Az SS05 módszer reprodukált eredményei `[szögletes zárójelbe rakott írógép stílusban]` láthatóak, a legjobb F-mértékek (%) pedig *dőlt betűvel* szedettek, valamint minden 94% fölötti F-mérték **félkövérrel** szedett. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

DOI:10.15774/PPKE.ITK.2018.007

	IOB1	IOB2	IOE1	IOE2	IOBES
T'n'T	[84,40]	[84,47]	[84,46]	[84,44]	[85,50]
NLTK T'n'T	84,64	84,70	84,70	84,74	85,64
PurePOS	85,47	85,52	85,50	85,52	86,11
HunTag3 Bigram	92,60	92,69	92,62	92,66	93,03
HunTag3 Trigram	92,83	92,84	92,84	92,81	93,17
HunTag3 CRFSuite	93,11	93,11	93,09	93,12	93,32
Hivatalos CRFSuite	*93,42*	*93,45*	*93,39*	*93,42*	*93,67*

A.4. táblázat. Megvizsgáltam minden címkézőt egyszerű többségi szavazással, **lexikalizáció nélkül**. Az SS05 módszer reprodukált eredményei [szögletes zárójelbe rakott írógép stílusban] láthatóak, a legjobb F-mértékek (%) pedig *dőlt betűvel* szedettek. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

	IOB1	IOB2	IOE1	IOE2	IOBES
T'n'T	88,58	88,65	88,63	88,59	89,27
NLTK T'n'T	88,65	88,72	88,72	88,68	89,36
PurePOS	89,19	89,23	89,23	89,26	89,77
HunTag3 Bigram					
HunTag3 Trigram					
HunTag3 CRFSuite	**94,15**	**94,17**	**94,14**	**94,18**	**94,51**
Hivatalos CRFSuite	***94,68***	***94,70***	***94,68***	***94,70***	***95,06***

A.5. táblázat. Megvizsgáltam minden címkézőt egyszerű többségi szavazással, **enyhe lexikalizációval**. A legjobb F-mértékek (%) *dőlt betűvel* szedettek és minden 94% fölötti F-mérték **félkövérrel** szedett. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

DOI:10.15774/PPKE.ITK.2018.007

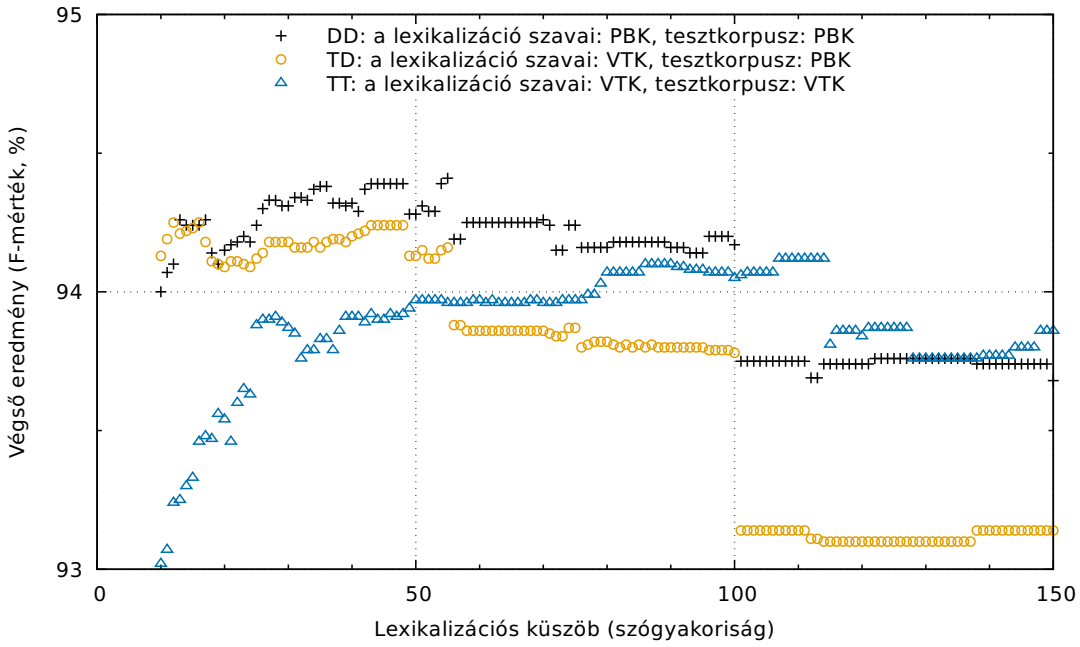
	IOB1	**IOB2**	**IOE1**	**IOE2**	**IOBES**
T'n'T	[91,73]	[91,74]	[91,73]	[91,74]	[92,18]
NLTK T'n'T	91,63	91,64	91,66	91,67	92,08
PurePOS	91,77	91,77	91,77	91,78	92,20
HunTag3 Bigram					
HunTag3 Trigram					
HunTag3 CRFSuite	93,76	93,75	93,74	93,75	93,96
Hivatalos CRFSuite	***94,33***	***94,32***	***94,31***	***94,33***	***94,65***

A.6. táblázat. Megvizsgáltam minden címkézőt egyszerű többségi szavazással, **teljes lexikalizációval**. Az SS05 módszer reprodukált eredményei [szögletes zárójelbe rakott írógép stílusban] láthatóak, a legjobb F-mértékek (%) pedig *dőlt betűvel* szedettek, valamint minden 94% fölötti F-mérték **félkövérrel** szedett. Minden reprezentáció esetén a *Hivatalos CRFSuite* konfiguráció teljesített a legjobban.

DOI:10.15774/PPKE.ITK.2018.007

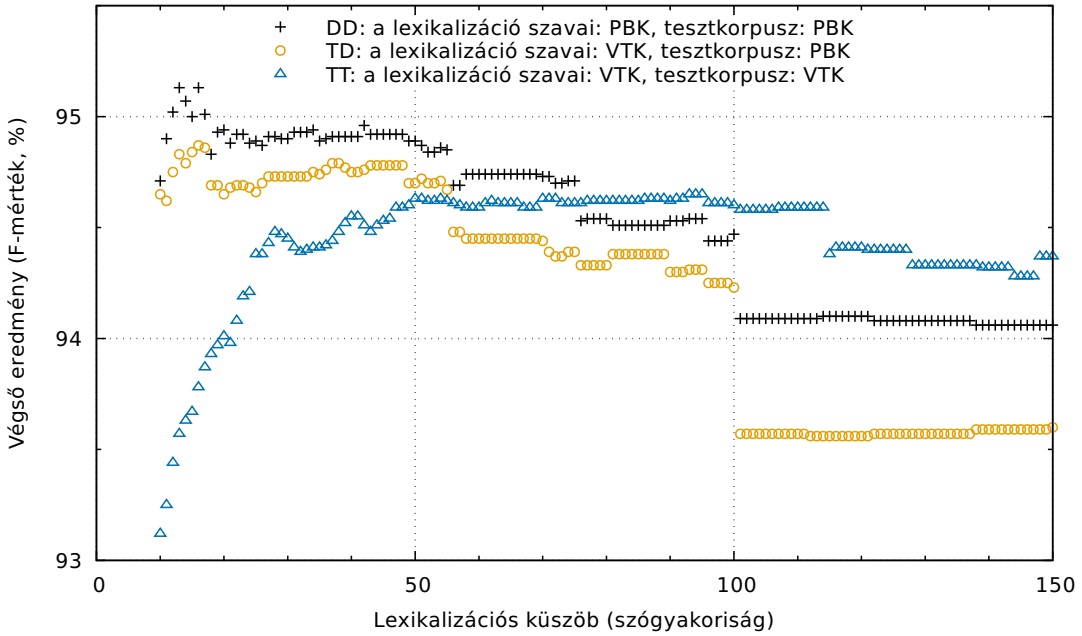
B. függelék

A küszöbérték és a teljesítmény aránya a többi reprezentáción



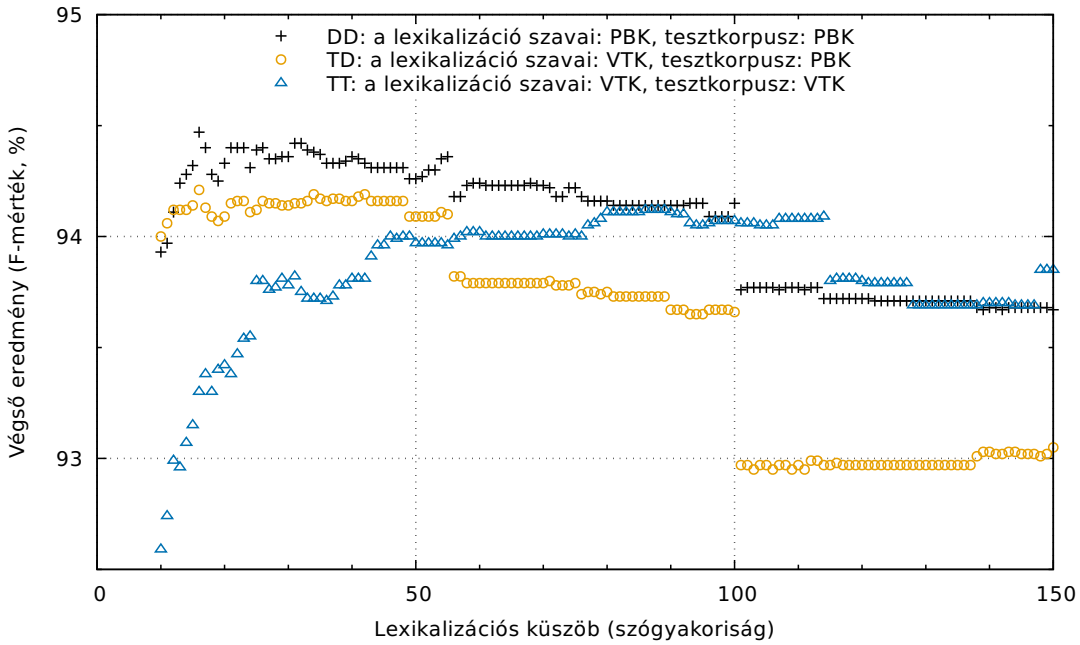
B.1. ábra. Lexikalizációs küszöbök az `IOB1` reprezentáción. Kb. 50-es küszöbértékig a teszthalmaz szavaival történt lexikalizáció jól láthatóan jobb eredményt ad. Továbbá látható egy szignifikáns esés a maximum elérése után 16-os küszöbértéknél.

DOI:10.15774/PPKE.ITK.2018.007



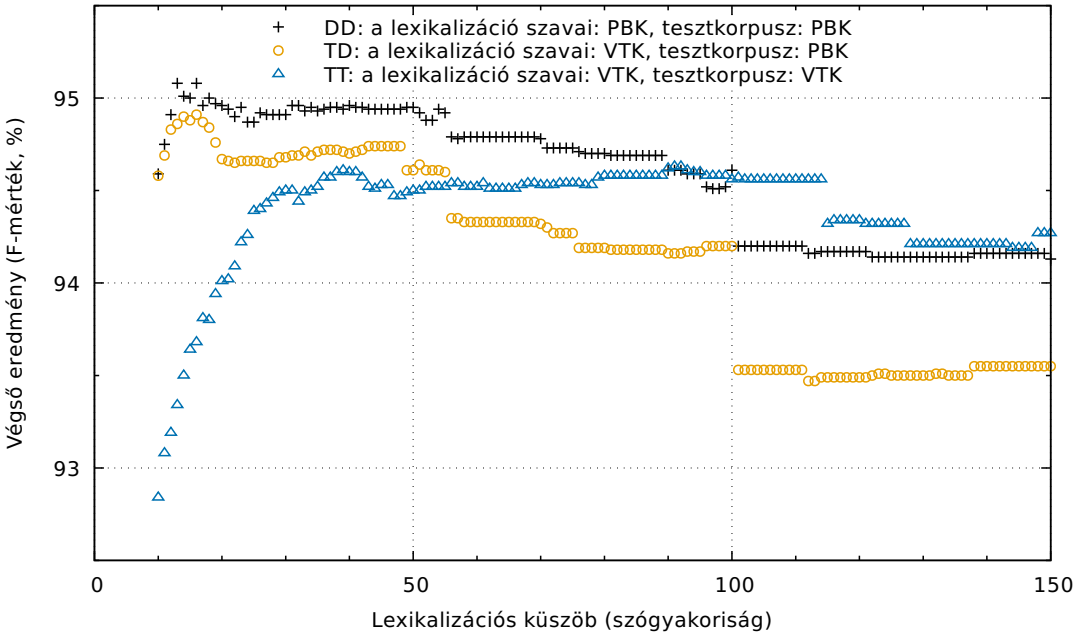
B.2. ábra. Lexikalizációs küszöbök az `IOB2` reprezentáción. Kb. 50-es küszöbértékig a teszthalmaz szavaival történt lexikalizáció jól láthatóan jobb eredményt ad. Továbbá látható egy szignifikáns esés a maximum elérése után 16-os küszöbértéknél.

DOI:10.15774/PPKE.ITK.2018.007



B.3. ábra. Lexikalizációs küszöbök az `IOE1` reprezentáción. Kb. 50-es küszöbértékig a teszthalmaz szavaival történt lexikalizáció jól láthatóan jobb eredményt ad. Továbbá látható egy szignifikáns esés a maximum elérése után 16-os küszöbértéknél.

DOI:10.15774/PPKE.ITK.2018.007



B.4. ábra. Lexikalizációs küszöbök az `IOE2` reprezentáción. Kb. 50-es küszöbértékig a teszthalmaz szavaival történt lexikalizáció jól láthatóan jobb eredményt ad. Továbbá látható egy szignifikáns esés a maximum elérése után 16-os küszöbértéknél.

DOI:10.15774/PPKE.ITK.2018.007

C. függelék

A lexikalizáció tulajdonságai különböző felosztásokon

Az alábbi táblázatok a közvetlen összetevős szerkezetek keresésének feladatát mutatják a CoNLL-2000 adathalmazon. Minden táblázat egy teszthalmazhoz tartozó, különböző tanító és paraméterállító halmazokat mutat eltérő módon lexikalizálva. A lexikalizációhoz használt szavak a `devel`-el jelölt oszlopok esetén a paraméterállításra használt halmazból származnak, míg a `test`-tel jelölt halmaz esetén a teszthalmazból. A *+szűrés*-sel jelölt oszlopokban csak a gyakori csoportokhoz tartozó szavakat vettem figyelembe, míg a *szűrés nélkül*-lel jelölt oszlopok esetén minden szót lexikalizáltam a referenciaadatra való tekintet nélkül. Az összes felosztás tekintetében az átlagos értékek az oszlopok sorrendjében a következők: 95,49 %, 93,48 %, 93,24 % és 95,17 %. Az értékekből látszik, hogy a *devel+szűrés* forgatókönyvet, mely az eredeti elképzelésnek felel meg, nem múlta felül egyik vizsgált módosítás sem.

DOI:10.15774/PPKE.ITK.2018.007

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**95,48**	93,55	92,99	95,21
2	**95,46**	93,63	92,94	95,21
3	**95,51**	93,40	92,68	95,01
4	**95,36**	93,44	92,53	95,07
5	**95,48**	93,39	92,63	95,18
6	**95,41**	93,38	92,67	95,10
7	**95,63**	93,29	92,49	95,15
8	**95,46**	93,55	92,93	95,25
9	**95,48**	93,14	92,35	95,07
10	**95,53**	93,52	92,53	95,04

C.1. táblázat. A teszthalmaz 1/1-es felosztása.

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**95,48**	93,55	92,99	95,21
2	**95,46**	93,63	92,94	95,21
3	**95,51**	93,40	92,68	95,01
4	**95,36**	93,44	92,53	95,07
5	**95,48**	93,39	92,63	95,18
6	**95,41**	93,38	92,67	95,10
7	**95,63**	93,29	92,49	95,15
8	**95,46**	93,55	92,93	95,25
9	**95,48**	93,14	92,35	95,07
10	**95,53**	93,52	92,53	95,04

C.2. táblázat. A teszthalmaz 1/2-es felosztása.

DOI:10.15774/PPKE.ITK.2018.007

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**95,46**	93,40	93,40	95,29
2	**95,48**	93,41	93,16	95,32
3	**95,51**	93,31	93,02	95,14
4	**95,30**	93,14	93,09	95,16
5	**95,47**	93,28	93,00	95,15
6	**95,29**	93,22	93,04	95,16
7	**95,68**	93,49	92,86	95,17
8	**95,43**	93,05	93,14	95,29
9	**95,43**	93,42	92,76	95,06
10	**95,55**	95,15	93,05	95,18

C.3. táblázat. A teszthalmaz 2/2-es felosztása.

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**95,20**	93,11	93,25	94,92
2	**95,00**	93,40	93,32	94,95
3	**95,05**	93,07	92,98	94,66
4	**95,08**	93,00	93,19	94,62
5	**95,14**	93,29	93,26	94,75
6	**95,12**	93,14	93,18	94,65
7	**95,21**	92,85	93,11	94,68
8	**95,18**	93,29	93,29	94,84
9	**95,08**	92,63	93,04	94,45
10	**95,15**	93,07	93,08	94,68

C.4. táblázat. A teszthalmaz 1/4-es felosztása.

DOI:10.15774/PPKE.ITK.2018.007

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**95,80**	94,30	94,19	95,47
2	**95,88**	94,33	93,78	95,40
3	**95,95**	94,12	93,75	95,57
4	**95,78**	94,15	93,67	95,65
5	**95,82**	94,03	93,81	95,77
6	**95,96**	93,83	93,84	95,49
7	**95,94**	93,87	93,78	95,76
8	**95,82**	93,94	93,68	95,38
9	**95,97**	93,81	93,74	95,60
10	**95,84**	94,16	93,83	95,35

C.5. táblázat. A teszthalmaz 2/4-es felosztása.

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**94,50**	92,99	92,92	94,24
2	**94,64**	92,99	92,82	94,22
3	**94,63**	92,67	92,59	94,29
4	**94,48**	92,92	92,61	94,33
5	**94,65**	92,49	92,47	94,41
6	**94,41**	92,72	92,80	94,34
7	**95,00**	92,65	92,50	94,35
8	**94,45**	92,93	92,78	94,42
9	**94,74**	92,49	92,32	94,04
10	**94,67**	92,95	92,86	94,31

C.6. táblázat. A teszthalmaz 3/4-es felosztása.

DOI:10.15774/PPKE.ITK.2018.007

felosztás	devel +szűrés	devel szűrés nélkül	test szűrés nélkül	test +szűrés (plafon)
1	**96,47**	93,83	93,95	96,04
2	**96,37**	93,85	94,05	96,18
3	**96,46**	93,77	94,02	96,11
4	**96,17**	93,72	93,97	96,26
5	**96,35**	93,82	94,11	96,26
6	**96,22**	93,87	94,01	95,93
7	**96,40**	93,82	93,96	96,04
8	**96,47**	94,10	94,07	96,06
9	**96,16**	93,65	94,04	96,29
10	**96,49**	93,92	94,07	96,18

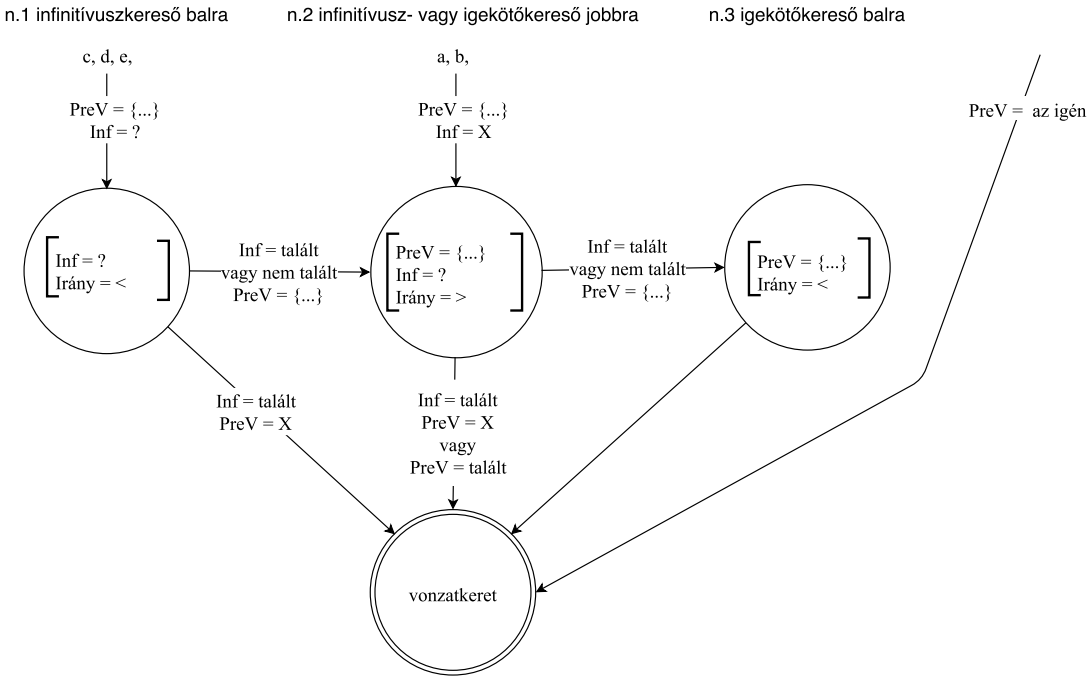
C.7. táblázat. A teszthalmaz 4/4-es felosztása.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

D. függelék

A *VFrame* keresőeljárás állapotainak automata reprezentációja



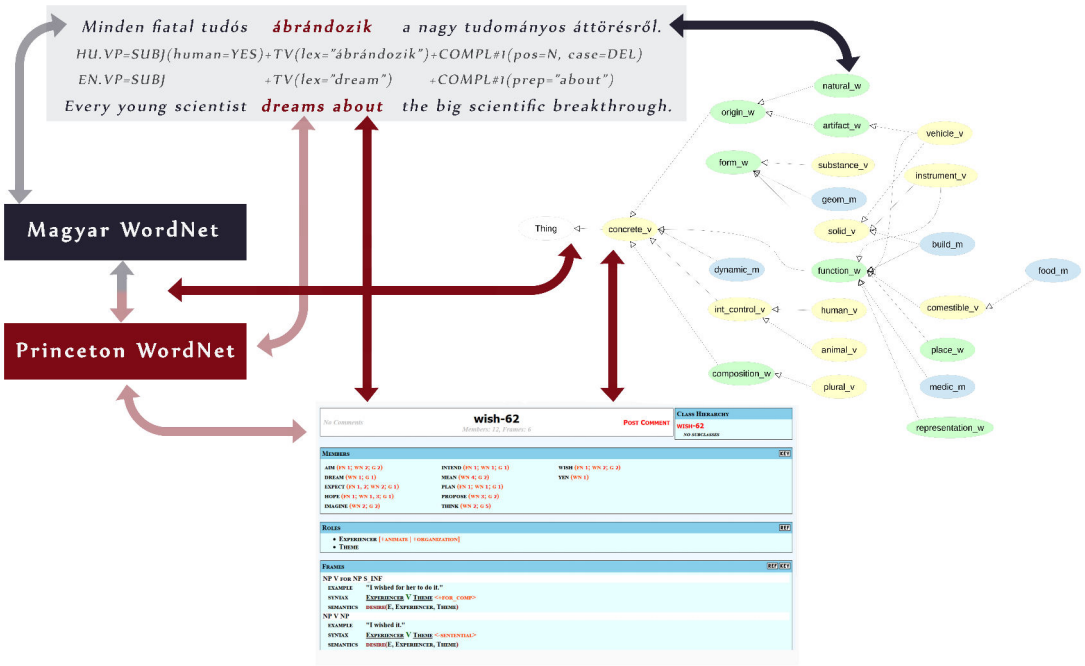
D.1. ábra. A *VFrame* keresőeljárás állapotainak véges állapotú automata reprezentációja (Vadász, Kalivoda és Indig 2017), amely lefedi az 5.9.1. fejezetben ismertetett öt igeosztályt.

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

E. függelék

A rendszerek összekapcsolásának vázlata



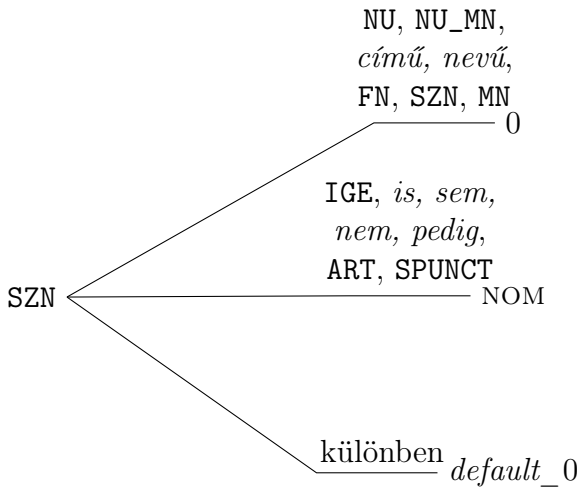
E.1. ábra. A MetaMorpho, a VerbIndex, a Princeton WordNet, a Magyar WordNet és az ontológiák kapcsolatai az *ábrándozik* vonzatkeretének tükrében (Indig, Miháltz és Simonyi 2016). (A bordó és sötétkék nyilak az általam létrehozott kapcsolatot jelölik, míg a halványbordó és a szürke nyilak a már meglévő kapcsolatokat.)

DOI:10.15774/PPKE.ITK.2018.007

DOI:10.15774/PPKE.ITK.2018.007

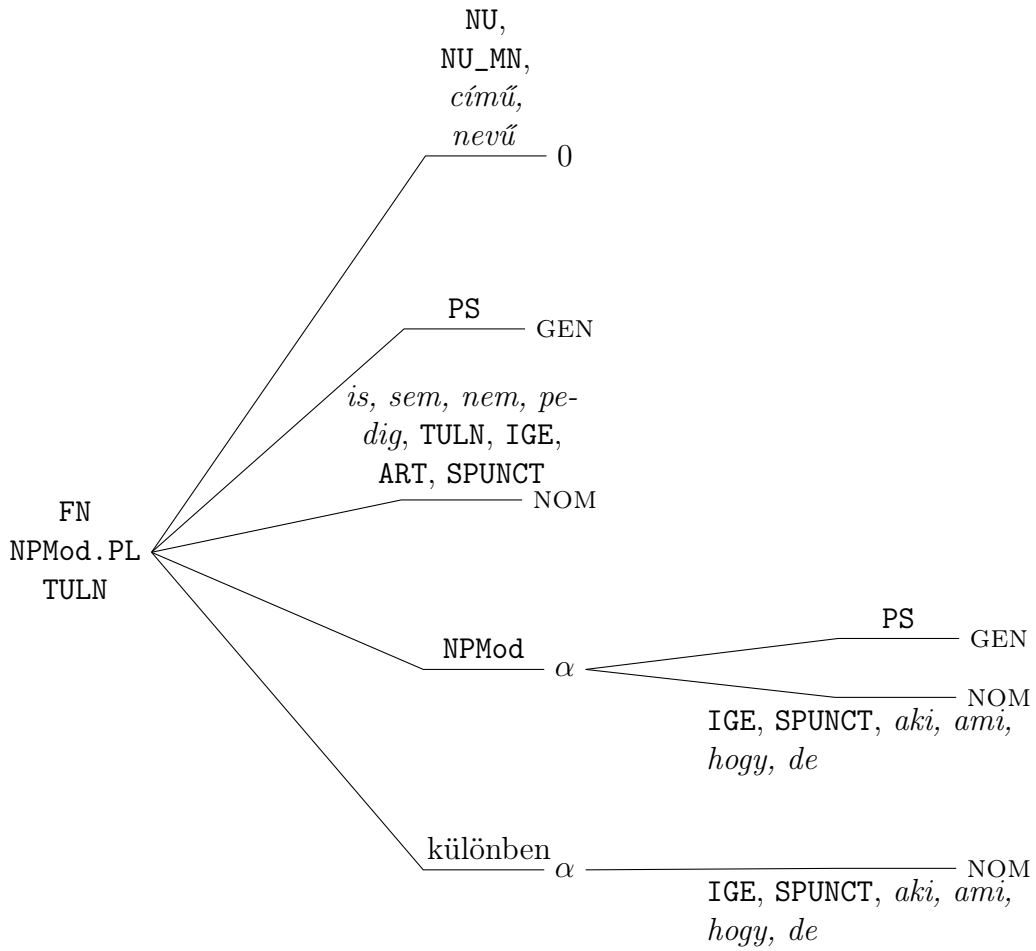
F. függelék

A `Nom-or-What` döntési fái



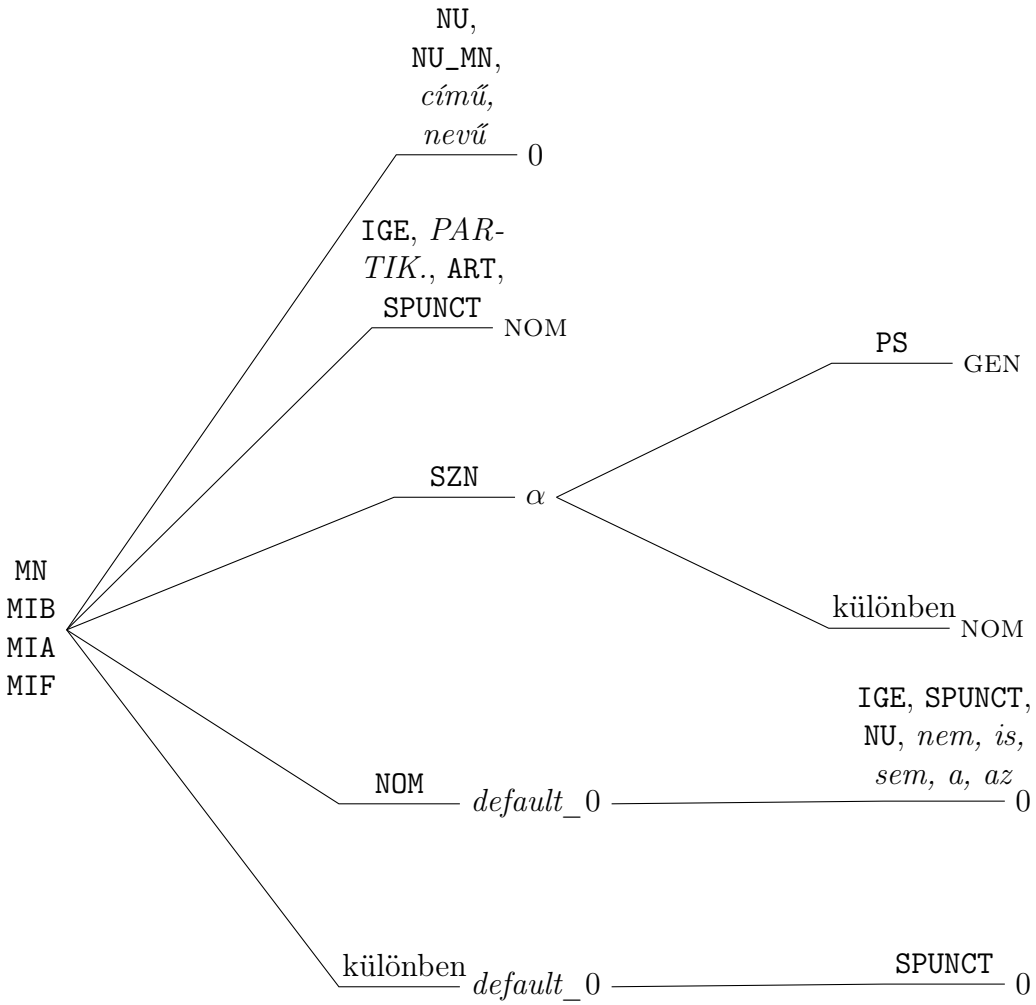
F.1. ábra. A számnevekre vonatkozó szabályok összefoglalása döntési fában. A fa gyökere az aktuális elem szófaji címkéje. A fa első szintjének élein az ablakban látható első elemen található információk szerepelnek.

DOI:10.15774/PPKE.ITK.2018.007



F.2. ábra. A főnevekre (köznevekre és tulajdonnevekre), illetve többesszámú melléknévnek, számnévnek, melléknévi igenévnek annotált elemekre vonatkozó szabályok összefoglalása döntési fában. A fa gyökere az aktuális elem szófaji címkéje. A fa első szintjének élein az ablakban látható első elemen található információk szerepelnek. A fa második szintjén lévő élek az ablak második elemén látható információkat tartalmazzák.

DOI:10.15774/PPKE.ITK.2018.007



F.3. ábra. A(z egyesszámú) melléknevekre és melléknévi igenevekre vonatkozó szabályok összefoglalása döntési fában. A fa gyökere az aktuális elem szófaji címkéje. A fa első szintjének élein az ablakban látható első elemen található információk szerepelnek. A fa második szintjén lévő élek az ablak második elemén látható információkat tartalmazzák. (A *PARTIK.* alatt az *is, sem, nem, pedig* szavakat értjük.)

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).