XV. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2019. január 24–25.

Konverterek magyar morfológiai címkekészletek között

Vadász Noémi, Simon Eszter

MTA Nyelvtudományi Intézet
E-mail: {vadasz.noemi, simon.eszter}@nytud.mta.hu

Kivonat A magyarra alkalmazott morfológiai annotációs sémák és címkekészletek sokszínűsége és eltérő dokumentáltsága ösztönzött minket abban a munkában, amelynek első lépéseit mutatja be ez a cikk. A munka két fő részből áll: egyrészt összegyűjtjük és közzétesszük a magyarra alkalmazott morfológiai annotációs sémákkal és címkekészletekkel kapcsolatos elérhető információkat, másrészt konvertereket írunk a címkekészletek között. Ebben a cikkben három konvertert ismertetünk.
Kulcsszavak: magyar nyelv, morfológia, annotáció, címkekészlet, konverzió

1. Bevezetés

Az elmúlt évtizedekben a magyar nyelvtechnológiai műhelyekben több morfológiai annotációs séma, valamint a hozzájuk tartozó kimeneti formalizmus és címkekészlet lett kifejlesztve. Közös vonásuk, hogy mindegyik a magyar nyelv morfológiáját kódolja, további számítógépes nyelvészeti feldolgozásra alkalmassá téve a szöveget. Olykor szükség van az egyes címkekészletek közötti konverzióra, például ha egy feldolgozó eszköz kimeneti formalizmusa nem egyezik meg egy következő feldolgozási lépés bemenetének formalizmusával. A konverzió egy plusz lépés beillesztése az elemzési láncba, így fennáll annak a veszélye, hogy nem várt hibák kerülnek a folyamatba. Ennek elkerülése érdekében törekedni kell a lehető legpontosabb konverzióra.

Kornai et al. (2004) [1] három fontos kritériumot támaszt, amelynek egy morfológiai elemző kimeneti formalizmusának meg kell felelnie: *informativitás*, *adekvátság* és *egyszerűség*. Az informativitás követelménye a címkekészletre vonatkozóan azt jelenti, hogy pontosan és a lehető legteljesebben tükrözze a szóalakban szereplő morfológiai információkat; az adekvátságé azt, hogy nyelvészetileg megalapozott kategóriákat tartalmazzon; az egyszerűségé pedig azt, hogy kézi és automatikus feldolgozásra is könnyen használható legyen. Ezek a kritériumok azonban gyakran ellentmondanak egymásnak, az ebből fakadó elméleti és formai különbségek nehezítik a címkekészletek közötti pontos konverziót.

A formai különbségek viszonylag könnyen áthidalhatók, azonban az elméleti különbségek már több problémát okoznak. Egyes annotációs sémák a szóalakban található összes morféma kódolására törekszenek, míg mások csupán az inflexiós

morfémákat kódolják. Eltérések lehetnek a szófajkészletben, bizonyos alkategóriák használatában, valamint az egyes nyelvi jelenségek kezelésének finomságában is. Az ideális cél a veszteségmentes konverzió, amihez a működő megoldást a leginkább közelíteni kell.

A használatban lévő morfológiai annotációs sémákat és címkekészleteket vizsgálva azzal szembesültünk, hogy sok esetben kevéssé dokumentáltak, valamint hogy a közöttük működő konverterek jellemzően csak saját, belső használatra készültek. Ezért a jelen cikkben ismertetett konvertereket nyílt forráskóddal és dokumentációval szabadon elérhetővé tesszük. A címkekészletek eltérő dokumentáltságát egy nyilvános GitHub repozitórium[1] létrehozásával orvosoljuk, amely tartalmazza az egyes annotációk által alkalmazott címkék teljes listáját, valamint az általunk fejlesztett konvertereket. A tárhely könnyen bővíthető más, eddig nem vizsgált vagy újonnan létrejövő címkekészletek ismertetésével, illetve az ezekre fejlesztett konverterekkel.

Jelen munkánkban először feltérképeztük a használatban lévő morfológiai címkekészleteket, erről lásd a 2. fejezetet. Emellett három konvertert készítettünk, amelyek a kurrens emMorph morfológiai elemző [2] kimeneti kódkészletét konvertálják egyrészt a magyarlánc 3.0 [3] által is használt Universal Dependencies (UD) kódkészletre, másrészt a magyarlánc 2.0 által is használt MSD-re, illetve annak egy jegy–érték párokban megfogalmazott verziójára, amelyre CoNLL-ként fogunk hivatkozni. A konvertereket a 3. fejezetben ismertetjük részletesen. A konverterek teljesítményét többféleképpen is kiértékeltük, amit a 4. fejezetben mutatunk be. A cikket összegzés és a jövőbeli tervek leírása zárja az 5. fejezetben.

2. Magyar morfológiai annotációs sémák

Ebben a fejezetben a jelenleg forgalomban levő magyar morfológiai annotációs sémákat ismertetjük – az általunk jelen fejlesztés kereteiben vizsgált formalizmusokra nagyobb hangsúlyt fektetve. Elsősorban azokra a formalizmusokra koncentrálunk, amelyek legalább egy széles körben használt és valamilyen formában elérhető korpuszban vagy egy hasonló tulajdonságokkal rendelkező elemző kimeneteként léteznek.

Az egyik ilyen annotáció az *MSD* (Morphosyntactic Description) [4], amely a magyarral együtt tíz nyelv részletes morfoszintaktikai reprezentációjára alkalmas. Különlegessége, hogy pozícióalapú kódolást valósít meg, vagyis a kód rögzített hosszúságú, és minden pozíciójához egy-egy morfoszintaktikai jegy van hozzárendelve, az egyes pozíciókat betöltő karakterek pedig a jegyekhez rendelt értékek. Az első pozíció mindig a szófaji kategóriáé, a többi pedig további morfoszintaktikai információkat kódol – például egy kijelentő módú, múlt idejű, egyes szám második személyű, tárgyas ragozású főige MSD-kódolásban így fest:

```
adtad ad Vmis2s---y
```

[1] https://github.com/dlt-rilmta/panmorph

Ez a szisztéma nem hierarchikus, vagyis nem tükrözi az egyes értékek közötti összefüggéseket, valamint a morfológiai jelöltséget sem, ám az alapos dokumentációból[2] kiderül, hogy melyek azok a kombinációk, amelyek előfordulhatnak az egyes címkékben, és melyek nem. Továbbá nem is sztringalapú, ami azt jelenti, hogy sem a lemma, sem a morfológiai szegmentumok, sem az allomorfok nem képezik részét a morfológiai elemzésnek. Nincsenek továbbá jelölve a derivációk sem, csak és kizárólag morfoszintaktikai kódok vannak.

A Szeged Korpusz és Treebank 1.0 [5] és 2.0 változata [6] MSD kódokat tartalmaz, valamint a magyarlánc 1.0 és 2.0 verziója is MSD kódokat adott ki. A magyarlánc 2.0-nak egy későbbi verziójában és a korpusz 2.5 változatában már a harmonizált MSD–KR kódkészlet található [7], amely néhány tulajdonságában eltér az eredeti MSD kódolástól. A továbbiakban erre a harmonizált változatra fogunk MSD-ként hivatkozni.

A Szeged Treebanknek létezik egy további verziója is, amely a 2009-es *Syntactic and Semantic Dependencies in Multiple Languages* című CoNLL shared task [8] követelményeinek megfelelő felépítésű – ezt hívjuk *CoNLL*-nek. Hangsúlyoznunk kell, hogy a CoNLL csak egy formátum, aminek a lényege, hogy a morfoszintaktikai információk linearizált jegy–érték párok formájában legyenek megfogalmazva, de az alkalmazott jegyek és lehetséges értékeik nem kötöttek. Ebben a változatban a CoNLL címkekészlet a Szeged Korpusz 2.0 MSD kódjaiból (tehát a még nem harmonizált MSD kódból) lett átkonvertálva.

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

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

A Szeged Dependency Treebanknek van egy olyan verziója is, amely a *UD* (Universal Dependencies and Morphology[3]) nevű nemzetközileg elterjedt, univerzálisnak szánt annotációs séma szabályait követi [9], valamint a magyarlánc 3.0 verziója is UD kódokat bocsát ki a morfológiai elemzés szintjén. A Szeged Dependency Treebank a UD 1. verziójának megfelelő címkéket tartalmazza. Azóta a UD 2. verziója is kijött már, de a magyar nyelvre és a Szeged Treebankre és így az azon alapuló eszközökre az újítások még nem lettek alkalmazva. A UD kódolás sokban hasonlít a CoNLL-hez: ez is egy linearizált jegy–érték struktúrát valósít meg, de itt a jegyek ábécésorrendben szerepelnek, és az értékkel nem kitöltött

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

jegyek nem jelennek meg. További tulajdonságaiban megegyezik a CoNLL fent ismertetett tulajdonságaival. A fenti példa ebben a kódolásban:

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

A legújabb magyar morfológiai elemző az *emMorph*[2], amely az e-magyar [10] szövegfeldolgozó eszközlánc morfológiai moduljaként is funkcionál. Ennek az elemzőnek az annotációs sémája jelentősen eltér az eddig ismertetettekétől, ugyanis sztringalapú, vagyis a lemma, a morfológiai szegmentumok és bizonyos esetekben az allomorfok is az elemzés részét képezik. További eltérést jelent, hogy nemcsak morfoszintaktikai információkat kódol, hanem olyan derivációkat is kezel, amelyeknek nem feltétlenül van köze az adott szó mondatbeli szerepéhez. Annyiban viszont hasonlít az MSD-hez, hogy nem hierarchikus, valamint nem tükrözi a morfológiai jelöltséget sem. Az emMorph többféle módon képes megjeleníteni a kimenetet aszerint, hogy tartalmazza-e a szóalakhoz rendelt tövet és a szegmentumokat a szófajcímke és az elemzések mellett. Mi a tövet és a morfémákat nem tartalmazó morfológiai kódot konvertáljuk. A fenti példa ebben a rendszerben[4] ábrázolva:

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

Léteznek még további magyar morfológiai annotációs sémák is, amelyeket megemlítünk, de jelen cikkben részletes leírást nem adunk róluk, ugyanis a fejlesztés jelenlegi fázisában még nem tudunk kész konvertereket kiállítani ezekre a formalizmusokra. Az egyik ilyen a *Humor*, illetve annak több változata [11,12,13]. A Humornak egy verziója lett használva az MNSZ2 [14] és egy másik verziója az Ómagyar Korpusz [15] építésénél is, ezért a későbbiekben tervezzük az ebből az irányból induló konverterek fejlesztését is. Egy másik formalizmus a *KR* kód [16], amelyet a hunmorph [17] morfológiai elemző bocsát ki, és amelyre a jövőben szintén tervezünk konvertereket írni.

3. A konverterek

Legyen szó bármilyen formátumok közti konverzióról, többféle megközelítés létezik. Az egyik, ha a bemeneti címkekészletről a kimenetire egy közvetlen leképezést valósítunk meg. Egy másik lehetséges módszer, ha – a gépi fordítás egy fajtájánál használt *interlinguá*hoz hasonlóan – egy köztes metaformátumot találunk ki, amire le tudunk képezni minden bemeneti formátumot, és amiből elő tudunk állítani minden kimeneti formátumot. Ez a magyar nyelv morfológiája esetében egy minden eddiginél részletesebb, a szokásos vitás kérdésekben (főnév vs. melléknév, inflexió vs. derivació stb.) kötelezően döntést hozó, a morfológiai annotációk fent felsorolt tulajdonságait (hierarchikusság, sztringalapúság stb.)

[4] A címkék feloldása példákkal együtt az e-magyar honlapján (`https://e-magyar.hu/hu/textmodules/emmorph_codelist`) található.

egyszerre birtokló újabb morfológiai annotációt eredményezne, ami lehetetlen vállalkozásnak tűnik. Ezért az első megközelítés mellett döntöttünk, és közvetlen leképezést csináltunk három irányba, ahol a bemeneti oldalon mindig az emMorph kódja áll.

Az emMorph címkekészletről történő konvertálásnak több előnye is van. Egyrészt az emMorph formalizmusa összességében részletesebb, mint a célformalizmusok, ezért a konverzió viszonylag kis veszteséggel megoldható. Másrészt pedig a magyar nyelvre készült kurrens elemzőláncba, az e-magyarba is az emMorph elemző van beépítve, így az e-magyarral elemzett szöveg tetszőlegesen átalakítható a kezelt címkekészletek valamelyikére a felhasználó céljainak megfelelően. Az `emmorph2msd` konverter kimenete a magyarlánc 2.0 által is előállított MSD kód; az `emmorph2conll` konverter kimenete a 2. fejezetben ismertetett, az MSD kód átalakításával kialakított jegy–érték struktúrájú CoNLL kód; az `emmorph2ud` konverter kimenete pedig a magyarlánc 3.0 által is előállított UD kód.

A konverterek kidolgozásához megvizsgáltunk néhány elérhető konvertert, azok működéséből, felépítéséből levontuk a számunkra fontos tanulságokat. Az egyik ilyen konverter az e-magyarban működő DepTool.java[5], amely az emDep modul számára konvertálja az emMorph címkéket a fent ismertetett CoNLL formátumra, de egy belső, kevert címkekészletet használva. A magyarláncban is több konverter működik a címkekészletek között (pl. a harmonizált MSD és a UD között[6]).

Az `emmorph2ud` konverter az e-magyar elemzőlánc legfrissebb, `emtsv` elnevezésű verziójában [18] kiváltotta a DepTool.java konvertert. Az elemzőláncba illeszkedve az `emMorph` kimenetét konvertálja az `emDep` modul számára fogyasztható jegy–érték struktúrájú UD címkékre, valamint kimeneti formalizmusként lehetővé teszi, hogy a felhasználók az eddig elérhető `emMorph` kimenet mellett UD morfológiai címkéket is kaphassanak.

A konverterek elkészítésekor akkor volt a legkönnyebb dolgunk, amikor egy-az-egyhez megfeleltetés állt fenn a bemeneti és a kimeneti oldal között. Ugyanakkor sok esetben szükség volt a címkék megfeleltetésekor aleseteket és kivételeket megfogalmazni. Ennek oka a konverterek közötti elméleti különbségekben keresendő. Szemléltető példaként tekintsük a szófajok és az azokat reprezentáló címkék esetét. Az emMorph formalizmusában a szófajokat ábrázoló címkék megkülönböztetett formát kaptak a morfológiai jegyekhez képest (`[/Adj]`). Ugyanakkor a melléknevekhez és határozószókhoz járuló felsőfokot kifejező morféma is a szófajcímkékhez hasonló formátummal rendelkezik (`[/Supl]`), így külön figyelmet kellett fordítanunk arra, hogy a felsőfokban álló melléknevek és határozószók szófaját kinyerjük. Ráadásul az emMorph a kimeneti címkekészletekkel ellentétben a derivációkat is megjeleníti a címkékben. A helyes konverzióhoz a legkülső képzett alak szófaját és az arra rakódó inflexiós jegyeket kellett kinyernünk az

[5] `https://github.com/dlt-rilmta/hunlp-GATE/blob/master/Lang_Hungarian/src/hu/nytud/gate/util/DepTool.java`

[6] `https://github.com/zsibritajanos/magyarlanc/blob/master/magyarlanc/src/main/java/hu/u_szeged/converter/univ/Msd2UnivMorph.java`

emMorph címkéből, és ezeket a jegyeket kellett a kimeneti címkekészletek megfelelő jegyeire konvertálnunk.

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

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

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

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

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

segédigeként is, ennek a kérdésnek az eldöntését a szintaxis területére toljuk, és csak egy igei címkét alkalmazunk.

A konvertereket `Python3`-ban implementáltuk. A kódok szabadon elérhetőek és felhasználhatóak GNU GPLv3 licenc alatt, míg a kódkészleteket ismertető dokumentációt és táblázatokat CC-BY-SA-4.0 licenc alatt publikáljuk a `https://github.com/dlt-rilmta/panmorph` repozitóriumban.

4. Kiértékelés

A konverterek teljesítményét több mérőszámmal is szemléltetjük. A kiértékeléskor igyekeztük valóban a konverzió minőségét megítélni, azonban a címkekészletek alapvető elvi különbségei, valamint a címkekészletekkel dolgozó elemző eszközök eltérő minősége is okozhatnak hibapontokat az egyes címkék összevetésekor.

A három konverter fejlesztése és kiértékelése hasonló módon zajlott. Először létrehoztuk a fejlesztéshez és a teszteléshez szükséges elemzéseket. A címkekészletek dokumentációi alapján elkészítettük a konverterek első verzióját, majd azzal átkonvertáltuk a fejlesztőanyagban található összes emMorph címkét UD, MSD, illetve CoNLL címkére. A kimenetben szereplő hibatípusokat elemeztük, majd a feltárt hibák alapján javítottunk a konverteren. Végül a tesztanyagon kiértékeltük a konverterek teljesítményét.

4.1. `emmorph2msd` és `emmorph2ud`

Mind az emMorph, mind az MSD és a UD címke produktívan előállítható, előbbi az emMorph elemző, utóbbi a magyarlánc valamely verziójának kimeneteként, ezért az `emmorph2ud` és az `emmorph2msd` fejlesztéséhez is korlátlan mennyiségű elemzést tudtunk előállítani. A fejlesztéshez a Szeged Treebankből kinyert összes szóalakot használtuk, amely összesen 152 056 tokent tesz ki.

A fejlesztéshez a tokeneket leelemeztük az emMorph-fal, amely 195 416 elemzést eredményezett, majd ezeket az elemzéseket konvertáltuk UD és MSD kódra. A tokeneket a magyarlánc 2.0-val és 3.0-val is[7] megelemeztük – ezek számítottak a gold standard adatnak, amelyhez a konverter kimenetét hasonlítottuk.

A konverterek tesztelésekor nem az egyes tokenek számítanak egy tesztesetnek, hanem a token és egy hozzá tartozó emMorph elemzés. Ennek megfelelően a fejlesztőanyagban annyi teszteset van, ahány emMorph elemzés (195 416). Ez azt is jelenti, hogy azokban az esetekben, amikor az emMorph hibás elemzést ad egy szónak – úgy is, hogy mellette esetleg jó elemzést is ad, ami egy másik tesztesetet képez –, de a magyarlánc összes elemzése között nem szerepel egy ugyanolyan jelentésű hibás elemzés, akkor olyan hiba is a konverter rovására íródik, amely nem a konverzió, hanem az emMorph hibája.

[7] Bár a Szeged Treebank elérhető mind emMorph címkékkel, mind UD és MSD címkékkel, mi mégis az újraelemzés mellett döntöttünk. Egyrészt a Szeged Treebankben alkalmazott konverzió és a kézi javítás eredményezte esetleges formai hibákat akartuk ilyen módon kiküszöbölni, másrészt így több teszteset áll a rendelkezésünkre.

A két konverter végső kiértékelését egy másik teszthalmazon végeztük, amelyhez a Webcorpus 100 000 leggyakoribb szavának listáját használtuk fel [19]. Ezekkel a fent leírtak szerint jártunk el, vagyis a szavakat megelemeztük az emMorph-fal, valamint a magyarlánc 2.0 és 3.0 verzióival is. Mivel a konverterek az emMorph címkék konvertálását vállalják, a fejlesztő és a tesztadatból kivettük azokat a szavakat is, amelyekhez az emMorph nem tudott címkét rendelni (a kimenet `'None'` volt). Voltak olyan szavak is, amelyekkel egyik elemző sem birkózott meg. Jellemzően ezek a tokenek az elemző számára valamilyen speciális jelentéssel bíró karaktert tartalmaztak (pl. * karakterre végződtek) – ezekből összesen 6 388 darab volt. A végső tesztanyag 93 606 tokenjéből az emMorph elemzést követően 120 714 címke állt elő, amelyből kivettük a `'None'` címkéket, így összesen 105 545 tesztesetünk maradt a kiértékelés elvégzésére.

4.2. `emmorph2conll`

A magyarlánc 2.0 előállít ugyan CoNLL címkéket, de csak a szintaktikai elemzés előkészítő lépéseként, a már morfológiai egyértelműsítésen átesett MSD címke átalakításával. Ez azt jelenti, hogy egy tokenhez nem az összes lehetséges elemzés CoNLL címkéje áll a rendelkezésünkre, hanem minden tokenhez csak egy. Éppen ezért az `emmorph2conll` esetében a Szeged Treebank hasonló annotációval ellátott változatára támaszkodtunk a fejlesztéskor és a teszteléskor is. Legelső lépésként felosztottuk a Szeged Treebankből kinyert szólistát (152 056 token) két részre olyan arányban, ahogy a másik konverternél aránylott egymáshoz a Webcorpusból és a Szeged Treebankből kinyert fejlesztő- és tesztelőanyag mérete. Így a fejlesztésre 94 245 token állt rendelkezésünkre, amely az emMorph-fal megelemezve 120 714 címkét eredményezett. A végső tesztelésre 57 781 token maradt, a `'None'` címkék kivétele után összesen 74 702 teszteset állt rendelkezésünkre. A kiértékelés során ugyanazt a három tesztet végeztük el, mint a másik két konverter esetében.

Az `emmorph2conll` esetében szintén egy token és egy emMorph címke párosa képez egy tesztesetet, ugyanakkor azt sem szabad elfelejteni, hogy a teszteléskor a tokenekhez nem az összes elképzelhető elemzés áll rendelkezésre, hanem csak azok az egyértelműsített jelentések, amelyek valóban előfordultak a tesztanyagban.

4.3. A mérések

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

Háromféle tesztet végeztünk el. Az első – legmegengedőbb – teszt során azt ellenőriztük, hogy a konvertált címke előfordult-e valaha a magyarlánccal elemzett tesztanyagban (tehát sem a tokent, sem az emMorph címkét nem párosítottuk hozzá). Bár feltételezhetjük, hogy a tesztanyag ugyan nem tartalmazza az összes

elképzelhető UD és MSD címkét, de a leggyakoribbakat biztosan, így ez a teszt annak a mérésére alkalmas, hogy valid címke jött-e létre a konverzió után. Vagyis ez csupán egy validitási kritériumot ellenőriz, önmagában nem elég mutatója a konverzió minőségének, elsősorban a fejlesztés során volt hasznos.

A második teszt volt a legszigorúbb, minden token esetében az ahhoz a tokenhez tartozó magyarlánc elemzésekkel vetettük össze a konvertált címkét. Emögött a mérőszám mögött az a feltételezés áll, hogy a kétféle elemző kimenetében szereplő címkék páronként megfeleltethetők egymásnak, mert ugyanaz a jelentésük. A valóságban azonban a két elemző sok jelenséget egészen eltérően kezel az annotációs sémák közötti elméleti különbségek miatt. Ráadásul az elemzők hibákat is vétenek, ami szintén nehezíti az összehasonlítást. Ezzel a szigorú mérőszámmal tehát nem pusztán a konverziót értékeljük ki, hanem a kétféle elemző különbségeit is kidomborítjuk, mert olyan esetek is hibásnak számítanak, amelyek a kétféle elemző eltérő minőségéből vagy megközelítéséből adódnak. Ezeket a hibákat nem válogattuk szét, így az eredményeket ennek tudatában kell értékelni.

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

4.4. Eredmények és diszkusszió

Az első teszt tehát azt vizsgálta, hogy valid címkék jönnek-e létre a konverzió során. Az 1. táblázatban látható, hogy mindhárom konverter nagyon magas eredményeket ért el ezen a teszten, ám ez a magas szám alapvető elvárás, amely egy konverterrel szemben támasztható. Magyarázatra szorul azonban a tény, hogy egyik konverterrel sem sikerült elérni 100%-os eredményt. Mindhártom konverter esetében átnéztük a nem validnak ítélt címkék listáját, és ellenőriztük, hogy a rendelkezésünkre álló dokumentációk alapján hibásak-e. A leírások alapján megállapítottuk, hogy a nem validnak ítélt címkék valójában validak, csak egyszerűen hiányoztak a gold standard adatból.

A 2. táblázatban ismertetett eredmények a második tesztre vonatkoznak, így az elvárásoknak megfelelően ezek a leggyengébbek. A 4.3. fejezetben ismertetett kiinduló ötlet alapján ez lenne a megfelelő mérés a konverzió minőségére, ám szem előtt kell tartani a tesztek során tapasztalt torzító hatást, amelyet az egyes címkekészletek és az elemzők közötti alapvető elméleti különbségek okoznak. Gyakori például, hogy az egyik eszköz csak melléknévi, míg a másik csak főnévi címkét ad egy szónak. Még ha a többi jegyet sikeresen konvertálja is a konverter, és a konverzió valójában helyes kimeneti címkét eredményezett, amiatt, hogy az ennek megfelelő címke hiányzik a gold standard adatból, a konverzió

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

1. táblázat. A konverterek eredményei az első teszten.

is hibásnak számít. Ez a probléma akkor merül fel, ha az egyes emMorph elemzésekhez nem párosítható elemzés az összes magyarlánc kimenet közül, tehát amikor a magyarlánc fedése kisebb.

	összes	TP	TN	Acc
`emmorph2ud`	105 545	87 506	18 039	82,91%
`emmorph2msd`	105 545	77 422	28 123	73,35%
`emmorph2conll`	74 702	52 176	22 526	69,85%

2. táblázat. A konverterek eredményei a második teszten.

Egy jellemző példa az anaforikus birtokos egyes és többes számú jelének előfordulása a fejlesztőanyagokban. A 3. táblázatból kiolvasható, hogy az emMorph szívesebben ad `[AnP]` és `[AnP.Pl]` címkéket a névszóknak, mint a magyarlánc különböző verziói. Természetesen az `emmorph2conll` kiértékelésekor ez a probléma fokozódik, mivel ott nem az összes lehetséges magyarlánc elemzés áll a rendelkezésünkre, hanem minden szóalakhoz csak egyetlen, a korpuszban lévő egyértelműsített elemzés.

	anaforikus birtokosok
emMorph	8 959
MSD	1 136
UD	5 804

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

A harmadik tesztet tekintjük a legalkalmasabb mutatónak a konverzió minőségére vonatkozóan. Az eredményeket a 4. táblázat ismerteti. Az `emmorph2ud` és az `emmorph2msd` konverterek esetében 97% fölötti eredményt értünk el, az `emmorph2conll` azonban jóval gyengébben, bár 90% fölött teljesített a teszten.

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

4. táblázat. A konverterek eredményei a harmadik teszten.

Azt feltételezzük, hogy az `emmorph2conll` gyenge eredményének az oka a kiértékelés módszerében keresendő. Míg az egyes tokenekhez az emMorph többféle elemzést is eredményezhetett, addig az annotált korpuszban egy tokenhez természetesen jóval kevesebb elemzés tartozott. Az 5. táblázat a Szeged Korpusz szólistájának token/címke arányát mutatja az emMorph-fal és a magyarlánc 3.0 verziójával megelemezve, valamint a CoNLL címkékkel annotált korpuszban. Minél magasabb ez a szám, annál több gold standard címkével tudjuk összevetni a konvertált címkét. Ez azt jelenti, hogy a harmadik teszt eredményét az `emmorph2conll` konverter kiértékelése esetében hasonló fenntartásokkal kell kezelni, mint a második teszt eredményeit.

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

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

A különböző morfológiai címkekészletek közötti konverzió során felmerül az eltérő tövesítés problémája is. Az emMorph mint derivációt is kezelő morfológiai elemző nyilván más tövet fog megállapítani egy képzett szó esetében, mint azok az elemzők, amelyek csak az inflexiós jegyeket kódolják. A tesztanyagokon kimértük, hogy az esetek mekkora részében jelenik meg az eltérő tövesítés. Ha egy szóhoz akár a bemeneti, akár a kimeneti oldalon több elemzés társul a tesztanyagban, akkor nehézkes a tövesítés összevetése. Ezért közvetlenül nem tudjuk összehasonlítani az elemzőket, csak közvetett módon. Azoknál a szavaknál hasonlítottuk össze a töveket, ahol a fenti kiértékelés alapján a 2. tesztben a konverter hibátlanul konvertált a címkék között.

A 6. táblázatban látható eredmények azt mutatják, hogy mindhárom címkekészletpárt tekintve az esetek legnagyobb részében nem különböznek a tövek a helyesnek ítélt konverziók között. Természetesen ez az eredmény nem jelenti azt, hogy a morfológiai kódok közötti konverziókor nem kell foglalkozni az eltérő tövesítéssel, az itt ismertetett konvertálók azonban egyelőre csak a morfológiai címkék közötti átváltást vállalják.

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

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

5. Összegzés

A konverterek elkészítésével lehetővé tesszük, hogy bárki könnyedén átalakíthassa az e-magyar elemzőlánc vagy az emMorph morfológiai elemző kimenetét egy általa választott morfológiai annotációs sémának megfelelően. Egyelőre az itt ismertetett három kódra tudunk konvertálni, de a jövőben tervezzük más be- és kimeneti kódkészletek közötti konverterek írását is. A fent bemutatott `emmorph2ud` konverter az e-magyar elemzőlánc új változatába is bekerült, ahol egyrészt egy közbülső láncszemként az emMorph kimenetét konvertálja az emDep modul számára fogyasztható jegy–érték struktúrájú UD címkére, másrészt pedig kimeneti formalizmusként lehetővé teszi, hogy az e-magyar elemzőláncot használók az eddig elérhető emMorph kimenet mellett UD címkéket is kaphassanak. Fontosnak tartjuk kiemelni, hogy az általunk készített konverterek forráskódja és a címkekészletek leírása szabadon elérhető a `https://github.com/dlt-rilmta/panmorph` nyilvános GitHub repozitóriumon keresztül.

Hivatkozások

1. Kornai, A., Rebrus, P., Vajda, P., Halácsy, P., Rung, A., Trón, V.: Általános célú morfológiai elemző kimeneti formalizmusa. In: II. Magyar Számítógépes Nyelvészeti Konferencia. (2004)
2. Novák, A., Siklósi, B., Oravecz, Cs.: A New Integrated Open-source Morphological Analyzer for Hungarian. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA) (2016)
3. Zsibrita, J., Farkas, R., Vincze, V.: A Toolkit for Morphological and Dependency Parsing of Hungarian. In: International Conference on Recent Advances in Natural Language Processing, Shoumen, Bulgária, INCOMA Ltd. (2013) 763–771
4. Erjavec, T.: MULTEXT-East Morphosyntactic Specifications. Version 3.0. (2004) http://nl.ijs.si/ME/Vault/V3/msd/html/.
5. Csendes, D., Csirik, J., Gyimóthy, T.: The Szeged Corpus: A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus. In Sojka, P., Kopecek, I., Pala, K., eds.: Text, Speech and Dialogue. Volume 3206 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2004) 41–47
6. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Proceedings of the 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, Springer (2005) 123–131

7. Vincze, V., Varga, V., Simkó, K.I., Zsibrita, J., Nagy, A., Farkas, R.: Morfológiai újítások a Szeged Korpusz 2.5-ben. In Tanács, A., Viktor, V., Veronika, V., eds.: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2014) 332–338
8. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štepánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task, Association for Computational Linguistics (2009) 1–18
9. Vincze, V., Farkas, R., Simkó, K.I., Szántó, Zs., Varga, V.: Univerzális dependencia és morfológia magyar nyelvre. In Tanács, A., Viktor, V., Veronika, V., eds.: XII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Tudományegyetem Informatikai Tanszékcsoport (2016) 322–329
10. Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: Az `e-magyar` digitális nyelvfeldolgozó rendszer. In Vincze, V., ed.: XIII. Magyar Számítógépes Nyelvészeti Konferencia. (2017) 49–60
11. Prószéky, G., Kis, B.: A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. ACL '99, Stroudsburg, PA, USA, Association for Computational Linguistics (1999) 261–268
12. Novák, A.: Milyen a jó Humor? In: I. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2003) 138–144
13. Novák, A.: A Humor új Fo(r)mája. In: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2014) 303–308
14. Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Proceedings of LREC 2014. (2014)
15. Simon, E.: Corpus Building from Old Hungarian Codices. In É. Kiss, K., ed.: The Evolution of Functional Left Perpiheries in Hungarian Syntax. Oxford University Press (2014) 224–236
16. Rebrus, P., Kornai, A., Varga, D.: Egy általános célú morfológiai annotáció. Általános Nyelvészeti Tanulmányok **XXIV.** (2012) 47–80
17. Trón, V., Kornai, A., Gyepesi, Gy., Németh, L., Halácsy, P., Varga, D.: Hunmorph: Open Source Word Analysis. In: Proceedings of the Workshop on Software, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 77–85
18. Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` – Egy formátum mind felett (2019) Jelen kötetben.
19. Kornai, A., Halácsy, P., Nagy, V., Oravecz, Cs., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In: Proceedings of the 2nd International Workshop on Web as Corpus. (2006)