Mennyiségből minőséget: Nyelvtechnológiai kihívások és tanulságok az MNSz új változatának elkészítésében

Oravecz Csaba, Sass Bálint, Váradi Tamás

MTA Nyelvtudományi Intézet
e-mail:{oravecz.csaba,sass.balint,varadi.tamas}@nytud.mta.hu

Kivonat A Magyar Nemzeti Szövegtár egymilliárd szavas új változatának fejlesztése során egyrészt a szövegek mennyiségéből, másrészt a nyelvi elemzés minőségével kapcsolatos elvárásokból adódóan számos olyan feldolgozási kérdés merült fel, melyekre nem lehetett a jelenleg hozzáférhető nyelvi elemző eszközök „polcról levett" alkalmazásával kielégítő választ adni. A tanulmány azokat a megoldásokat és javaslatokat mutatja be, melyek hozzájárulnak ahhoz, hogy az olyan jelentős méretű korpuszokban is, ahol a manuális hibajavítás nem lehetséges, az annotáció minősége megfeleljen a felhasználói elvárásoknak.

Kulcsszavak: korpusz-előfeldolgozás, tokenizálás, morfológiai elemzés, szófaji egyértelműsítés

1. Bevezetés

Nagy méretű szövegkorpuszok előállításakor két kritikus dimenzió határozza meg a fejlesztés körülményeit: a mennyiség és a minőség. Az utóbbi időben mindkét irányban jelentős előrelépések történtek, egyrészt részletes, „mély" elemzést tartalmazó szöveggyűjtemények jelentek meg, másrészt szinte mindennapossá vált a milliárd szavas méret [1,2,3,4]. A két követelmény között nem kézenfekvő az ideális kompromisszum, amit egy további fontos tényező is nagyban befolyásol, a korpusz majdani felhasználóinak igényei. Az MNSz éppen ebből a szempontból speciális helyzetű, egyszerre kívánja kiszolgálni a számítógépes alkalmazásokat, a nyelvészeti kutatásokat és a nyelv iránt érdeklődő nagyközönséget is. Ennek következtében az új változat elkészítésekor számos kihívással kellett szembenézni, amire a peremfeltételekhez legjobban illeszkedő megoldásra volt szükség. A dolgozat azokat a fejlesztés során felmerült problémákat és javasolt megoldásokat ismerteti részletesen, melyek véleményünk szerint tanulságosak és hasznos információval szolgálnak azok számára, akik magyar nyelvi szövegeket nagy mennyiségben kívánnak nyelvtechnológia eszközökkel feldolgozni. A Szövegtár fejlesztésének általános kérdéseit megelőzően már tárgyalja [5] és [6] is. A jelen tanulmány azonban ezeken túlmutatva azokat a problémákat fejti ki részletesebben és más hangsúlyokat meghatározva, melyek leginkább érdeklődésre számítanak a magyar kutatóközösség körében.

2. Előfeldolgozás, normalizálás

A korpuszépítés egyik fontos lépése a beszerzett forrásszövegek előszűrése, feldolgozása addig a célszerűen sztenderd formátumig, amely alkalmas arra, hogy a nyelvi elemző rendszer bemeneteként szolgáljon. Az új MNSz ezzel kapcsolatos munkálatai alapvetően a szokásos, a régi szövegtárban is (részben) elvégzett feladatokat jelentették (forrásszöveg kivonása, nyelvazonosítás, duplikátumok eltávolítása, bekezdés szintig kódolt XML formátumra alakítás stb.), ezért itt csupán azt a problémát tárgyaljuk részletesebben, amely részben gyakorlati, kényszerű szempontok miatt új kihívást jelentett.

Az elektronikus szövegek túlnyomó része manapság UTF-8 karakterkódolású. Az Unicode szabvány által rendelkezésre bocsájtott szimbólumhalmaz tág teret ad azonban azoknak a típusú „visszaéléseknek", ahol egyes szövegek, szövegrészek valamilyen megjelenítési, formázási okból nem kanonikus karaktereket használnak. Ez a típusú információ, gyakorlatilag procedurális *markup*, ideális esetben természetesen leválasztandó, elkülönítendő a kanonikus tartalomtól. Az MNSz esetében ezt a fajta megjelenítési információt nem őrizzük meg. Nem egyértelmű azonban, hogy az Unicode szabvány által meghatározott normalizáló algoritmusok közül bármelyik is közvetlenül alkalmazható lenne. A szövegek változatossága és mennyisége nem teszi lehetővé, hogy minden egyes szövegegységre megvizsgáljuk, hogy vajon elegendő-e, ha a kanonikus ekvivalencia elvének megfelelő normalizáló formát választunk, vagy fennállnak-e a feltételei annak, hogy az esetenként a karakterek szemantikáját is befolyásoló, enyhébb kompatibilitás ekvivalenciát biztosító formát alkalmazzuk [7].

Az alkalmazott adatvezérelt megoldást végül az is jelentős mértékben meghatározta, hogy a 3.1. részben részletezett okok miatt a szövegek további átalakítására, egy ISO-8859-2 kódolásra történő konverzióra is szükség volt. Ennek folyamán az ISO-8859-2 kódtáblán kívüli karakterek szabványos XML numerikus entitásokra képződtek le. Ezek gyakorisági listáját vetettük alá egy manuális vizsgálatnak, melynek segítségével kialakítottunk egy olyan egyedi leképezést, amely a benne szereplő entitásokat, amennyiben lehetséges, a nyelvi szempontból ekvivalens absztrakt karakter ISO-8859-2 kanonikus alakjára képezi le. Ez a leképezés magában foglalja az Unicode kódtábla azon teljes tartományait, melyek a vizsgálat alapján előszeretettel használatosak procedurális markupként (pl. teljes- vagy félszélességű karakterek), függetlenül attól, hogy minden elemük konkrétan előfordult-e a szövegekben, illetve a leképezhető XML néventitásokat is (1. ábra). Az ezen kívül eső karakterek maradtak XML numerikus entitások, ily módon az eredeti UTF-8 szövegek a korpusz szempontjából releváns információ elvesztése nélkül voltak konvertálhatók.[1]

[1] Ez véleményünk szerint célravezetőbb, mint valamely sztenderd transzliterációs megoldás (például az *iconv* segédprogram TRANSLIT opcióval) alkalmazása, amely esetében lényegi információ (pl. ékezet) veszhet el az átalakítás során.

```
Entitás    Latin2 karakter
&#x01c4;        DŽ
&#x01c5;        Dž
&#x01c6;        dž
&#x01ca;        NJ
&#x01cb;        Nj
&#x01cc;        nj
...
&Uacute;        Ú
&Ucirc;         Ű
&Udblac;        Ű
&Uuml;          Ü
&Yacute;        Ý
```

1. ábra. Normalizáló leképezés részlet.

3. Elemzés és annotáció

Mind a korpusz méretéből, mind a leginkább a felhasználói igények által képviselt minőségi kényszerből természetszerűleg adódnak feldolgozási nehézségek a nyelvi elemzés minden szintjén. Ez megköveteli olyan elemző eszközök használatát, melyek rugalmasak, robusztusak és jól testre szabhatók, egyedi igényekre alakíthatók. Az annotált korpuszokban az elemzés magáért az elemzésért van, ez alapvetően különbözik attól, amikor valamilyen további alkalmazásban van az eszközök kimenete beágyazva. Mások a követelmények, a kiértékelés alapja pedig az elemzés minősége, és nem egy befogadó alkalmazás teljesítménye. Itt elkerülhetetlenül merül fel az a kérdés, hogy van-e készen kapható, „polcról levehető" és a célnak megfelelő magyar nyelvtechnológiai elemzőkészlet, illetve milyen mértékben használhatók egyes komponensek az adott feladat elvárt szintű megoldására.

A továbbiakban három szokványos alapvető elemzési lépést vizsgálunk: a tokenizálást/szegmentálást, a morfológiai elemzést és a szófaji egyértelműsítést. Mindhárom esetben meghatározzuk azokat a kívánalmakat, amelyeket a bevezetőben említett peremfeltételek mellett a felhasználandó eszköznek teljesítenie kell, megvizsgáljuk, hogy a rendelkezésre álló eszközök mennyiben felelnek meg ezeknek a kívánalmaknak, ismertetjük az általunk alkalmazott, néhány esetben a praktikus kényszer által is vezérelt megoldást, illetve esetenként javaslatot teszünk olyan fejlesztési lépésekre, amelyeket feltétlen szükségesnek tartunk ahhoz, hogy egy adott elemzési feladatot magas minőségben megoldani képes, konfigurálható és tárgykörre adaptálható eszköz jöjjön létre. Mivel a jelen dolgozat leginkább egy helyzetjelentés, és az ezzel a helyzettel szembesülve, jelentős részben pragmatikus szempontok által indokolt megoldások kiválasztási módszereinek a leírása, vagyis semmiképp sem tekinthető a tárgyalt eszközök sztenderd környezetben történő minőségi kiértékelési jelentésének.

3.1. Mondatszegmentálás, tokenizálás

A tokenizálás és mondatszegmentálás esetében a minőségi kényszer a magas *pontosság* mellett magas *fedést* is megkíván, vagyis meglehetősen változatos (szépirodalomtól a webes blogokig) szövegtípusokban kell jó eredményt elérni, nemcsak a rendkívül változatos konfigurációban előforduló mondatok határainak megállapításában, hanem speciális szövegelemek (nyílt tokenosztályok, például URL-ek, email címek stb.) felismerésében is. Minőségi magasabb szintű nyelvi annotációhoz elengedhetetlenül szükséges a pontos tokenizálás, amely messze túlmutat az egyszerű reguláris kifejezésekre támaszkodó, alkalmi szkriptek által nyújtott megoldási lehetőségeken [8].

Mint sok más nyelvfeldolgozó alkalmazásban, itt is két megközelítés szokásos, statisztikai modell [9,10] alapú illetve szimbolikus, szabályalapú [11]. Az előbbi típusban elérhető megoldások a szövegmennyiség és változatosság miatt széles körű tanítás, tesztelés és finomhangolás nélkül nem teljesíthetnek kielégítően, az ehhez szükséges idő és erőforrás viszont a projekt során nem állt rendelkezésre, így az elterjedtebb és számos készen használható eszközt kínáló utóbbi megközelítés volt kézenfekvő.

1. táblázat. Tokenizálók tulajdonságai.

	MtSeg[2]	Europarl[3]	huntoken[4]	Nagel[5]	FreeLing[6]	NYTI lánc[7]
1.	+	+	+	+	+	+
2.	C	perl	flex/C	C	C++	perl
3.	latin1/2	UTF-8	latin1/2	latin/UTF-8	UTF-8	latin2
4.	minimális	minimális	jó	minimális	kevés	nincs
5.	kérdéses	jó	jó	jó	közepes	jó
6.	+ + + + +	+ + +	+	++	+ + ++	+ + +
7.	nyelvi modulok konfigurálhatóság	egyszerű	nyelvi tudás	fejleszthető	gyors	egyszerű
8.	elavult kód, lassú	nincs nyelvi tudás	tokenizálási hibák	nincs nyelvi tudás	nincs nyelvi tudás	nincs nyelvi tudás

1.: forrás elérhető 2.: implementáció 3.: kezelt karakterkódolás 4.: dokumentáció 5.: fejleszthetőség 6.: becsült fejlesztési igény 7.: előnyök 8.: hátrányok

[2] Már nem elérhető, saját példány.
[3] http://search.cpan.org/~achimru/Lingua-Sentence-1.03/lib/Lingua/Sentence.pm
[4] http://mokk.bme.hu/resources/huntoken
[5] http://www.cis.uni-muenchen.de/~wastl/misc/tokenizer.tgz
[6] http://www.lsi.upc.edu/~nlp/freeling
[7] Marcia Munoz mondatrabontója (http://aye.comp.nus.edu.sg/~forecite/services/uiuc-srl/srl-demo2/bin/sentence-boundary2.pl) és a Grefenstette-féle tokenizáló szkript (http://nora.hd.uib.no/corpora/1999-3/0348.html) házi használatra módosított változatban

A MNSz tekintetében a legfontosabb szempontok az adatmennyiség miatt a gyorsaság, a komplex nyelvi elemek (nyílt tokenosztályok) felismerhetősége miatt a nyelvi tudás, a szövegek változatossága miatt pedig a doménilleszthetőség lehetősége voltak. Az 1. táblázat foglalja össze egy informális vizsgálat alapján néhány tipikus eszköz jellemző tulajdonságait főként az említett szempontok alapján. Az „egyszerűség" annyiban előny, hogy gyors és robusztus működést eredményez, abban pedig természetesen hátrány, hogy a (teljesen) hiányzó nyelvi tudás beépítése rendkívül erőforrásigényes. A vizsgálat egyértelmű tanulsága, hogy azon túl, hogy több eszköz csak szerény mértékben képes a triviális szegmentálási/tokenizálási megoldásnál[8] többet nyújtani, nincs minden szempontnak megfelelő, készen használható eszköz magyar nyelvre, amely az UTF kódolást is képes lenne kezelni. Ennek következtében olyan kompromisszumot kellett kialakítani, amely a leggazdaságosabb eredményt adja a befektetett fejlesztés – kimeneti minőség tekintetében.

A választás a komplex tokenek beépített kezelési képessége és a nyelvi illesztés miatt a *huntoken* eszközre esett, és ez a projekt keretében visszafordíthatatlan elkötelezettséget jelentett, az elemzés alatt felmerülő problémákkal szembesülve a kiindulópontra visszatérni és egy újabb eszközzel ismét elölről kezdeni a szükséges fejlesztést és kiegészítést nem volt lehetséges. Első lépésben a karakterkódolási problémát kellett megoldani, a 2. részben tárgyalt módon. Ezen kívül a szövegek sokfélesége felszínre hozott számos implementációs hibát, szabályhiányosságot, ezeket javítani kellett. A módosítás több száz sornyi új kódot, többek között a rövidítések kezelési módjának teljes átdolgozását, és a kimeneti formátum egyszerűsítését eredményezte, és végül messze túlment az eredetileg becsült minimális fejlesztési igényen. Eredményül viszont a működési körülményekhez képest a legjobb minőségű elemzést adó eszköz jött létre. Ez azonban mégsem tekinthető egy magyar nyelvi tokenizáló/szegmentáló eszközre adott optimális megoldásnak. Nem teljesülnek ugyanis azok a feltételek, melyek ehhez szükségesek lennének.

Az UTF kódolás natív kezelése természetes követelmény, de ennél alapvetőbb, architekturális hiányosságok is felmerülnek, nemcsak ennek, hanem sok más eszköznek az esetében is. Az egy lépésben történő elemzés veszélye, hogy a fedés növelésével együttjáró egyre összetettebb szabályrendszert rendkívül nehezen lehet karbantartani, a komplex kifejezések által meghatározott halmaz elemei a humán fejlesztő számára már nem láthatók át teljes körűen, így a halmaz tartalmazhat olyan elemeket, melyek más kifejezések nyelvébe is beletartoznak. Ha ilyenkor a szabályok közötti hierarchia nincs egyértelműen és konfigurálhatóan meghatározva, akkor az alkalmazott implementáció belső szabályrendezése ér-

[8] Triviális megoldásnak tekintjük a szóköz(értékű) karakterek és központozás mentén történő tokenizálást és az általában mondatzáró központozás utáni nagybetűs elem által meghatározott mondathatár bejelölését, esetenként kiegészítve segédlexikon alapú rövidítésdetektálással.

vényesül, ami hibához vezethet.[9] Erre a problémára jó megoldás a tokenizálási feladat többlépcsős megközelítése. A mondatra bontás és a tokenizálás elkülönítése gyakorlatilag magától értetődő, de az utóbbit is célszerű felbontani: az első lépésben azonosított elemi egységek további lépésekben alkothatnak összetett egységeket, és minden lépésért különálló, egyedileg konfigurálható modul felel.[10] A sebességcsökkenés megtérül a pontosabb működésen. Ennek a felépítésnek a legjobb példája a Multext Segmenter [13,14], ez azonban az elavult implementáció miatt nem használható, az alkalmazott architektúra és az elérhető funkcionalitás viszont jó kiindulópont.

A doménilleszthetőség olyan funkció, ami elengedhetetlenül szükséges egy robusztus és változatos bemenetet kezelni képes eszköz esetén. Egyszerű illusztrációja ennek például a szóközhiány mondatvégi központozásjelek után és nagybetű előtt, hiszen előfordulnak olyan szövegtípusok, ahol ez az esetek nagy részében hiba (mindennapi prózai szövegek), de olyanok is, ahol viszont általában nem hiba (számítógépes szakszövegek). Ezért az adott szövegtípushoz kell illeszteni bizonyos szabályok alkalmazhatóságát, és ezt egyszerűen konfigurálhatóvá kell tenni.

Összegzésképpen legcélszerűbbnek látjuk az alapoktól felépíteni a fenti követelményeknek megfelelő rendszert, szintetizálva mindazt a felhalmozott tudást és előnyös tulajdonságot, ami a jelenleg rendelkezésre álló rendszerekben megtalálható.

3.2. Morfológiai elemzés

A magyar nyelvi korpuszokban szokásos gyakorlat, hogy a morfológiai annotáció az utolsó képzőt magában foglaló és ez által meghatározott szófajú tövet és az ehhez járuló inflexiós toldalékolást tartalmazza. Felmerült az igény azonban a morfofonológiai kutatások kiszolgálása érdekében, hogy a morfológiai elemző kimenetéből további hasznos és kutatási kérdésekhez könnyen lekérdezhető információt is szolgáltassunk, és most először a morféma (és fonéma) szintű elemzés és annotáció is hozzáférhető legyen a korpuszban[11]. Ez mind az alkalmazott eszközzel, mind a kapott elemzés feldolgozásával kapcsolatban teljesen új problémákat vetett fel, különösen a morfémaszintű strukturális többértelműségek és szóösszetételi anomáliák feloldásában. Ez a követelmény az eszközválasztást is jelentősen befolyásolta, tekintve, hogy azon magyar nyelvi elemzők közül, amelyek a morfémákra történő felbontást is visszaadják kimenetként az egyik (*Xerox*) teljesen zárt és jelen körülmények között megváltoztathatatlan rendszer, így sem hibajavításra sem bővítésre nem ad lehetőséget, a másik (*ocamorph*) túlgenerálása olyan mértékű, amit rendkívül körülményes kezelni, és az ilyen részletes

[9] Esetünkben például a `flex` belső szabályhierarchiáját felülírni csak rendkívül körülményesen lehet [12], ami az alkalmazott szabálymennyiség mellett karbantarthatatlan.

[10] Az általunk használt eszköz alapvetően csak a rövidítéseket próbálja így kezelni.

[11] Ezen túlmenően egyes képzők jelenléte a nyelvi elemzés magasabb szintjein is releváns információ lehet.

elemzést kiadó üzemmódban sebességben is elmarad a végül általunk felhasznált harmadik (*HUMOR*) eszköztől.

A morfémákra bontás strukturális többértelműségeinek feloldására az [5]-ben említett egyszerű heurisztika (a legrészletesebb felbontás a választott elemzés) alkalmazásának érdekében minden a fent említett módon számított tő+inflexió alakhoz hozzárendeljük a lehetséges derivációs elemzések halmazát (2. ábra). Az

```
[kenyérkalaposság/FN.PSt2.SUB]:{
kenyérkalap[FN]+os[_SKEP]+ság[_PROP]+otok[PSt2]+ra[SUB];
kenyérkalap[FN]+os[_SFN]+ság[_COL]+otok[PSt2]+ra[SUB];
kenyér[MN]+kalap[FN]+os[_SKEP]+ság[_PROP]+otok[PSt2]+ra[SUB];
kenyér[MN]+kalap[FN]+os[_SFN]+ság[_COL]+otok[PSt2]+ra[SUB]
}
```

2. ábra. Elemzési alternatívák.

egyes halmazokon belül tehát vesszük a legrészletesebb (legtöbb morfémát tartalmazó) elemzési uta(ka)t, és minden egyes morfémára eltároljuk a lehetséges elemzéseit, az esetleges többértelműségeket megőrizve. Ugyancsak tároljuk az összetételek minden elemét (3. ábra). Azt a meglehetős kihívást jelentő kérdést, hogy az elemzés ezen szintjén keletkező többértelműségek automatikus feloldása miként lenne lehetséges, a projekt keretében nem vizsgáltuk. A végső annotáció tartalmaz még egy további egyszerű reprezentációt a morfémahatárok megjelölésére (`kenyér+kalap+os+ság+otok+ra, bokr+os+od+ás`).

```
[kenyérkalaposság/FN.PSt2.SUB] -> {
 stem  => {[kenyér],[kalap]},
 'os'  => {SKEP,SFN},
 'ság' => {PROP,COL},
 'otok'=> {PSt2},
 'ra'  => {SUB}
}
```

3. ábra. Morfémaszintű reprezentáció.

A szóösszetételek nagyfokú produktivitását kezelni képes elemző elkerülhetetlenül túlgenerál, ennek automatikus szűrése olyan eddig nem kielégítően kezelt probléma [15], aminek a megoldását a projekt nem tudta felvállalni. Erre az esetre ismét egy adatvezérelt megközelítés volt az alkalmazott eljárás alapja, ahol egy gyakorisági lista morfológiai elemzése után az összetételnek elemzett alakok közül a leggyakoribbak, illetve bizonyos tipikus utótagra[12] végződők teljes körű

[12] Pl. *ad, árok, dám, dia, est, jak, kád, kan, kos, lak, tat, tó, velő.*

manuális vizsgálat alá estek. Ennek eredménye egy mintaillesztő szűrő erőforrás lett, jelenleg 112 mintát és több ezer szűrt szóalakot tartalmazva (4. ábra)

```
(ár|borz|fog|hal|láz|mar|rag|szak|tag)\[FN\]\+ad\[IGE\]
(ár|borz|fog|hal|láz|mar|rag|szak|tag)\[FN\]\+ad(ó|ás)\[FN\]
...
(Balla|bor|Bor|cella|Cella| ... szó|téboly|vaj|Ver|zár)\[FN\]\+dám\[FN\]
(abszorber|adapter|áttétel ... törvény|zsilip|zsindely)\[FN\]\+est\[FN\]
```

4. ábra. Hamis összetételeket (pl. *láz+adó, áttétel+est*) szűrő minták.

Mindezen újdonságnak számító hozzáadott információ ellenére a morfológiai elemzés kérdése az MNSz-ben sincs teljes körűen megoldva. Hiányzik a tokenizáló és a morfológiai elemző gördülékeny együttműködése például a tokenizáló által felismert komplex alakulatok toldalékolásának felismerésében, és magának az elemzőnek is maradtak hibái. Az eddig a morfológiai elemzés területén befektetett hazai erőfeszítések igazán hatékony kihasználását egyértelműen akadályozza egy közös, harmonizált (vagy legalább harmonizálható) kimeneti kódkészlet és reprezentáció hiánya a különböző morfológiai elemzőkben. Szükség lenne egységes és nyíltan hozzáférhető segéderrőforrások, lexikonok közösségi fejlesztésére is.[13] Összességében, ahogy a tokenizálásnál, itt is úgy látjuk, hogy az eddigi eszközök tudását szintetizáló szabadon elérhető, megfelelően gyors és testre szabható elemző létrehozása lenne a kívánatos megoldás.

3.3. Szófaji egyértelműsítés

A szófaji, morfoszintaktikai egyértelműsítés, mint a nyelvi elemzés egyik sarokpontja, már magyar nyelvre is viszonylag alaposan feltárt területnek számít, a nagyméretű korpusz és a szövegek változatossága azonban jelenthet még kihívást [1]. Egy ennél sokkal alapvetőbb problémát is érdemes azonban felvetni. Valóban teljes körűek az ismereteink az egyes eszközök teljesítményéről? Összehasonlíthatók-e egyáltalán a különböző rendszerek eredményei? Az eddig megjelent rendszerekről (lásd pl. többek között [16,17,18,19]) közölt teljesítményadatok alapján lényegében lehetetlen kijelenteni, hogy az egyik módszer jobb a másiknál, annyira eltérőek a kiértékelési környezetek, a korpusz kódkészlete, a felhasznált külső erőforrás (például a morfológiai elemző). Nem kizárható, hogy némi változás a kézi tanító korpuszban vagy a tagkészletben már nagyobb hatással van a végeredményre, mint az alkalmazott rendszer lecserélése egyikről a másikra [20]. Rendkívül fontos tehát egy sztenderdizált kiértékelési környezet és protokoll meghatározása, az egyértelműsítés hibáinak megalapozott vizsgálata és értékelése. Ez mindezidáig hiányzik a magyar szakirodalomból. Mivel az nehezen vitatható, hogy egy morfológiai elemző kimenetének integrálása az egyértelműsítési folyamatba jelentős javulást eredményez, először ezt az információforrást

[13] Egy ilyenre példa lehet az itt említett összetételi szűrő.

kellene egységesíteni, ennek hiányában nincs lehetőség objektívan értékelni. További problémát okoz a korpuszkód-készletek, illetve a bennük tárolt információ különbözősége. Hiába mérjük az egyes egyértelműsítő eszközök teljesítményét egységes kódkészlettel, részletesen vizsgálni kell azt is, hogy a kódkészlet nem elfogult-e valamelyik eszközzel szemben, vagyis pont azokat a jellemzőket tartalmazza, amik az egyik eszköznek hasznosak, míg olyan jellemzőket, amiket esetleg másik eszköz tudna felhasználni, nem tartalmaz, neutralizál. Ez a típusú kiértékelés igen munkaigényes, így az MNSz készítésének keretében erre nem volt lehetőség. Ezért a továbbiakban olyan általános alkalmi vizsgálatok eredményét mutatjuk be, amelyek ettől függetlenül is tudnak tanulsággal szolgálni.

Három egyértelműsítő eszköz [16,18,19] kimenetét vizsgáltunk, ebből kettő [16,19] azonos kódkészlettel tanítható volt, a harmadik [18] előre megépített modellel és kódkészlettel rendelkezett. A tévesztési mátrixok alapján jellemző idioszinkratikus és paradigmatikus hibákat lehetett azonosítani, az eszközök közötti minőségi különbség viszont a fenti problémák miatt megállapíthatatlan volt. A 2. táblázat olyan mindegyik eszközre jellemző[14] tévesztéseket mutat be, melyek egy-egy tipikus problémát és egyben megoldási lehetőséget is illusztrálnak.

2. táblázat. Jellemző tévesztések (aszimmetrikus mátrixból).

Helyes kód	Összes előf.	Ebből hibás	(%)	Hibás kód	Előfordulása	(%)
1. AS_V	446	80	(17.94%)	VS3PI	46	(57.50%)
2. AS_A	2655	74	(2.79%)	NS3NN	42	(56.76%)
3. NS3PN	350	48	(13.71%)	D__D	25	(52.08%)
4. R__P	306	24	(7.84%)	C	14	(58.33%)
5. VS3SD	24	11	(45.83%)	VS3RD	11	(100.00%)

AS_[AV]: mn/ige tövű mn.; VS3PI: ige, múlt i.;
NS3NN: fn. egyes szám nom.; NS3PN: 3.szem. egyes sz. fn.-i névmás;
D__D: névelő; R__P: hsz.; C: kötőszó;
VS3SD: ige, felszólító m.; VS3RD: ige, kijelentő m.

1. A melléknév(i igenév) és a múlt idejű ige megkülönböztetése olyan összetett információt igényel, ami nem érhető el az eszközök által épített modelleken maradéktalanul. Ilyen esetben célszerű külön modellt használni a feladatra.
2. A főnév és melléknév homonímák megkülönböztetése bizonyos esetekben a humán annotátorok számára sem egyértelmű, és elméleti nyelvészeti szempontból sem teljesen tisztázott terület. Meghatározott eseteket lehet automatikusan javítani[15], de ennél több nem nagyon várható.
3. Az ebben a típusban található hibák (*az* mint névmás illetve névelő) legegyszerűbb megoldása célzott modellel, akár külső szabállyal a legegyszerűbb.

[14] A mért értékek minimális eltéréssel megegyeznek mindhárom eszközre.
[15] Pl. névelő előtt legyen mindenképpen főnév a megoldás, de itt is gondot okozhatnak az elliptikus szerkezetek.

4. A kötőszó és határozószó (*így, amikor* stb.) elemzés megkülönböztetésének nehézsége hasonlít az 1. esethez, ezen túl bizonyos esetekben az is célravezető lehet, vagyis kevesebb hibát eredményez, ha a ritkább elemzést egyszerűen figyelmen kívül hagyjuk a modellben, tehát meg sem próbálunk egyszerű automatikus megoldást alkalmazni arra a problémára, ahol több hibát okoz a megoldás alkalmazása, mint amennyit meghagy a nem alkalmazása.
5. Ez a típus az 1. eset szófajon belüli megjelenése, hasonló konklúzióval.

Az MNSz-nek ebben a feldolgozási lépésében a fenti eredmények figyelembe vételével igazi opportunista döntés született és lényegében az eddig is használt saját célra alakított feldolgozó láncot használtuk, amely egy szabályokat használó előszűrőből és a morfológiai elemző kimenetével megszorított nagyon gyors HMM alapú egyértelműsítőből áll [21]. Nem volt egyértelmű bizonyíték arra, hogy létezik jelentősen jobb és hasonlóan gyors megoldás, ezért nem volt indokolt egy jól működő eszközlánc lecserélése.

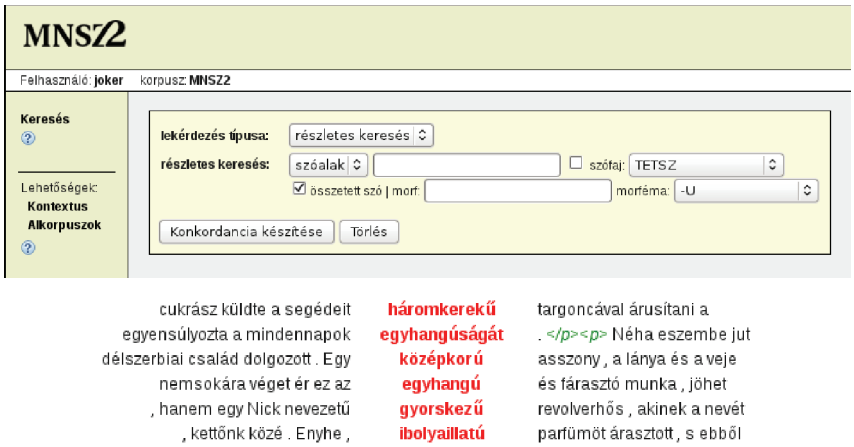
4. Korpuszkezelés és megjelenítés

Az MNSz új változata a megszokott `http://mnsz.nytud.hu` címen érhető el. A korpusz mögött egy korszerű, megbízható korpuszkezelő motor működik [22]. Sebessége a milliárd szavas méret mellett is megfelelő, a lekérdezőfelület válaszideje rövid.

A motorhoz tartozó felület eleve számos hasznos beépített funkciót tartalmaz, melyek újdonságot jelentenek az MNSz régi változatához képest. Nagy mennyiségű találat esetén is lekérhetjük az összes találatot, és kényelmes formátumban elmenthetjük további feldolgozásra. Egy gombnyomással testre szabhatjuk a megjelenítést, rendezhetjük a konkordanciát. A kapott találatokat újabb lekérdezéssel szűrhetjük, több lépésben is. Különféle gyakorisági listákat készíthetünk, és kollokációs vizsgálatot is végezhetünk.

Annak érdekében, hogy az egyes nyelvi szinteken megjelenő igen részletes elemzési információt felhasználóbarát módon hozzáférhetővé tegyük, az eredeti felületet számos ponton bővítettük. Egyrészt kiegészítettük a magyar inflexiós morfológia jelenségeit lefedő menürendszerrel, mely funkcionalitásában megfelel a régi MNSz-felület hasonló részének. Újdonság, hogy fonológiai jegyek, fonémaosztályok alapján is kereshetünk. Például a részletes keresőben beállított `{pal,aff}u.*{son}` kereséssel a palatális affrikátával kezdődő, *u*-val folytatódó és szonoránsra végződő szavakat keressük, és ennek megfelelően a találatokból képzett gyakorisági lista a *csupán*, *dzsungel*, *csupor* szavakkal kezdődik. Szintén újdonság, hogy a korpuszban meglévő morfémaszintű elemzésnek köszönhetően vizsgálhatók az összetett szavak, illetve hozzáférünk a derivatív morfológiához: az egyes morfémákhoz és konkrét morfmegvalósulásokhoz is (5. ábra).

A fentiek mellett arra is lehetőség van, hogy a rendszer belső korpuszlekérdező nyelvét – a CQL-t – közvetlenül használjuk, és általa rugalmasan hozzáférjünk a korpuszban rejlő információ egészéhez.



5. ábra. Az *-ú* képzőt tartalmazó összetett szavak lekérdezése és a válaszkonkordancia egy részlete.

5. Összefoglalás

A MNSz munkálatai során egyértelműen igazolódott, hogy a nyelvi elemzés egyes szintjein nincs, és persze nem is nagyon lehetséges olyan széles alkalmazhatóságú, elegendően gyors kész eszköz, amely magas minőségű megoldást képest adni a korpusz teljes szövegspektrumán. Ezért rendkívül fontos az eszközök konfigurálhatósága, a hatékony doménillesztés lehetősége. Ez viszont gyakorlatilag hiányzik minden jelenleg hozzáférhető szimbolikus alapú eszközből, a sztochasztikus megoldásokban pedig sztenderdizált nyelvi erőforrások híján jelentős befektetést kíván. A statisztikai modelleket alkalmazó eljárások hiába taníthatók az adott doménen, ha az elvárt pontosságú annotációhoz szükséges tanító adat nem áll rendelkezésre, és előállítása jelentős befektetést igényel, így nem lehet kijelenteni, hogy egyértelmű előnyben lennének a szimbolikus megoldásokkal szemben (lásd például a mondatszegmentálás és tokenizálás problémáját).

A minőségi igény következtében nagyon fontos szempont, hogy a nagy mennyiségű szöveg feldolgozása elkerülhetetlenül számos hibát derít fel az alkalmazott eszközben, és ezek folyamatos javításához elengedhetetlen az eszköz erőforrásaihoz történő átlátható hozzáférés, az alkalmazott modell(ek) gyors és rugalmas újraépítésének lehetősége.

Nagy szükség lenne végre egy közös fejlesztés eredményeként előálló magyar BLARKra [23], az eszközök, a hozzájuk szükséges erőforrások és az objektív, sztenderdizált tesztkörnyezet tekintetében is, ahol az egyes feldolgozási lépések akár többféle modullal is elvégezhetők, ezek azonban jól definiált API-n keresztül kommunikálhatnak egymással.

Hivatkozások

1. Wei-yun, M., Huang, C.R.: Uniform and effective tagging of a heterogeneous gigaword corpus. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006), Genoa, Italy (2006) 2182–2185
2. Parker, R., Graff, D., Kong, J., Chen, K., Maeda, K.: English Gigaword Fifth Edition. Linguistic Data Consortium (2011)
3. Halácsy, P., Kornai, A., Németh, P., Varga, D.: Parallel creation of gigaword corpora for medium density languages – an interim report. In: Proceedings of the International Conference on Language Resource and Evaluation (LREC08). (2008)
4. Ferraresi, A., Zanchetta, E., Baroni, M., Bernardini, S.: Introducing and evaluating ukWaC, a very large web-derived corpus of English. In: Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google. (2008) 47–54
5. Oravecz, Cs., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., szerk.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (2014)
6. Váradi, T., Oravecz, Cs.: A Magyar Nemzeti Szövegtár egymilliárd szavas új változata. Magyar Tudomány **175** (2014) 1054–1061
7. The Unicode Consortium: The Unicode Standard, version 7.0.0 (2014)
8. Forst, M., Kaplan, R.M.: The importance of precise tokenizing for deep grammars. In: Proceedings of Fifth Language Resource and Evaluation Conference (LREC2006), Genoa, Italy (2006)
9. Kiss, T., Strunk, J.: Unsupervised multilingual sentence boundary detection. Computational Linguistics **32** (2006) 485–525
10. Reynar, J.C., Ratnaparkhi, A.: A maximum entropy approach to identifying sentence boundaries. In: Proceedings of ANLP-97, Washington, USA (1997)
11. Grefenstette, G., Tapanainen, P.: What is a word, what is a sentence? problems of tokenization. In: Papers in Computational Lexicography. COMPLEX'94, Budapest, Research Institute for Linguistics (1994) 79–87
12. Flex Frequently Asked Questions: (Flex is not matching my patterns in the same order that i defined them) http://flex.sourceforge.net/manual/FAQ.html.
13. Ide, N., Véronis, J.: MULTEXT: Multilingual Text Tools and Corpora. In: Proceedings of the 15th Conference on Computational Linguistics. (1994) 588–592
14. Erjavec, T., Ide, N., Petkevic, V., Véronis, J., Schuman, A.R.: Multext-east: Multilingual text tools and corpora for central and eastern european languages. In: Proceedings of the TELRI (Trans-European Language Resources Infrastructure) European Seminar. (1995) 87–97
15. Novák, A., Pintér, T.: Milyen a még jobb Humor? In: Magyar Számítógépes Nyelvészeti Konferencia, Szeged (2006) 60–69
16. Oravecz, Cs., Dienes, P.: Efficient stochastic part of speech tagging for Hungarian. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas (2002) 710–717
17. Halácsy, P., Kornai, A., Csaba Oravecz: Hunpos – an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, ACL (2007)
18. Zsibrita, J., Vincze, V., Farkas, R.: magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In: Proceedings of RANLP. (2013) 763–771

19. Orosz, Gy., Novák, A.: Purepos 2.0: a hybrid tool for morphological disambiguation. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013), Hissar, Bulgaria (2013) 539–545
20. Manning, C.D.: Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In: Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I. CICLing'11, Berlin, Heidelberg, Springer-Verlag (2011) 171–189
21. Oravecz, Cs., Dienes, P.: Large scale morphosyntactic annotation of the Hungarian National Corpus. In Hollósi, B., Kiss-Gulyás, J., szerk.: Studies in Linguistics. Volume VI., Debrecen, Institute of English and American Studies, University of Debrecen (2002) 277–298
22. Rychlý, P.: Manatee/Bonito – a modular corpus manager. In: Proceedings of the 1st Workshop on Recent Advances in Slavonic Natural Language Processing, Brno: Masaryk University (2007) 65–70
23. Krauwer, S.: The Basic Language Resource Kit (BLARK) as the first milestone for the language resources roadmap. In: Proceedings of SPECOM, Moscow (2003)