ENDRÉDY ISTVÁN – NOVÁK ATTILA
MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport,
PPKE ITK
{endredy.istvan,novak.attila}@itk.ppke.hu

Szótövesítők összehasonlítása és alkalmazásaik

In this paper, we present a lemmatizer based on the Humor (High-speed Unification MORphology) morphological analyzer. Humor-compatible morphologies have been developed for many languages. The Hungarian Humor analyzer was integrated in various commercial products and used in many scientific projects. The lemmatizer, which we present in detail, is capable of handling and correctly lemmatizing all special morphological constructions in Hungarian, as well as those in other Humor-supported languages. We evaluate the performance of the Hungarian lemmatizer on 82.000 sentences of the Szeged Corpus, the biggest manually checked Hungarian annotated corpus, comparing its performance to that of five other stemmers. Our lemmatizer performs best with regard to most evaluation measures. In addition, we present two further applications of the tool, using and evaluating it as an automatic diacritic (accent) restoration system for accent-stripped input and in a tool that is capable of marking the mid *e* phoneme present in some Hungarian dialects, but not marked by standard Hungarian orthography.

Bevezetés

Az információkereső és lekérdezőrendszerek működése, illetve számos más szövegfeldolgozási feladat megoldása általában feltételez egy olyan algoritmust, amely képes a szövegekben szereplő szavak tövének vagy lehetséges töveinek megállapítására. Különösen érvényes ez a magyarhoz hasonlóan bonyolult morfológiájú nyelvek esetében, hiszen egy-egy lexémának a szövegekben előforduló rengeteg különböző toldalékolt alakja csak így képezhető le egy közös formára, amelynek segítségével a ragozott alakok mind megtalálhatóak. Az adott szóelőfordulás értelmét nem helyesen tükröző tő megadása hibás vagy hiányzó találatokhoz vezet. Így a tövesítés minősége befolyásolja az egész rendszer minőségét, annak ellenére, hogy az ilyen rendszerek általában többé-kevésbé kifinomult algoritmusokat alkalmaznak a találatként kapott dokumentumok relevancia szerinti sorba rendezésére, ami a tövesítés gyengeségeit részben elfedheti. A tövesítés kevésbé kritikus kérdés kevéssé ragozó nyelveknél, például az angolnál, a magyar esetében azonban a sok toldalékolt alak miatt egy gyenge minőségű tövesítő sokat ronthat a keresőrendszer hatékonyságán.

A különböző szótő-megállapító algoritmusokat több szempontból csoportosíthatjuk. Az egyik lényeges szempont az, hogy az adott algoritmus által visszaadott tő vagy tövek mennyire felelnek meg az adott nyelv lexikográfiai hagyományai szerint szótőnek tekinthető alakoknak. Lemmatizálónak nevezzük azokat az eszközöket, amelyek mindig a szótárírói hagyománynak megfelelő reprezentáns alakot, a lemmát adják

vissza. A magyar esetében ez névszóknál általában az egyes szám alanyesetű alakot, igéknél a jelen idő kijelentő mód egyes szám harmadik személyű alakot jelenti, de más nyelvek esetében az utóbbi helyett gyakran inkább az infinitívusz használatos. A lemmával szemben ráadásul általában elvárás, hogy létező szóalak legyen. Ezért ha egy adott defektív paradigmájú lexéma paradigmájából éppen a szokásos lemmának megfelelő alak hiányzik (pl. a *megsínyli* ige esetében), akkor másik alakot választanak. Általánosabban tövesítőnek (stemmer) nevezünk minden olyan eszközt, amely valamilyen tőalakot létrehoz az adott szóalakhoz, bármiféle megszorítás nélkül arra nézve, hogy az adott alak valóban az adott lexéma paradigmájának valamelyik tagja legyen, vagy akár csak létező szóalak legyen az adott nyelven. Az általános, szótárt nem tartalmazó gyors algoritmikus tövesítő algoritmusok a szóalak csonkolásával gyakran ilyen tőalakokat hoznak létre, amely azért bizonyos feladatokra elegendő is lehet.

Ebben a cikkben egy általunk készített magyar nyelvű lemmatizálót mutatunk be, amelynek teljesítményét összevetjük más, a magyar nyelvre alkalmazható lemmatizáló- és általános tövesítőeszközök teljesítményével. Az elkészített modul számos cég alkalmazásaiba beépült: erre épül a Microsoft Indexing Service, az Országos Atomenergetikai Hivatalban tárolt dokumentumok tárolására és keresésére szolgáló rendszer, az MTI szerkesztőségi rendszere, a PolyMeta kereső. Az MTA Nyelvtudományi Intézete által készített Magyar Nemzeti Szövegtár második bővített kiadásának (MNSZ2) morfológiai annotálása ugyancsak ezzel az eszközzel készült.

A Humor szóelemző

A magyarhoz hasonló agglutináló nyelvek esetében nehézkes lenne a lehetséges szóalakok milliárdjait felsorolni, és így eltárolni (igék esetén egy szó lehetséges releváns alakjainak száma a képzők produktivitása miatt akár az ezres nagyságrendet is elérheti). Ezért a lehetséges szóalakok nyelvtanon alapuló definiálása nyújthat hatékony megoldást a szóelemzés problémájára.

Jelen megoldás az unifikációalapú modellt használó Humor (High-speed Unification MORphology) szóelemzőre (Prószéky és Kis 1999), illetve az ahhoz készült magyar szóalaktani adatbázisra épül (Novák 2003).

A Humor elemző a szavakat morfokra (minimális szóelemek, morfémák konkrét lehetséges alakjaira) bontja, miközben ellenőrzi a szomszédos morfémák kapcsolatait, illetve a teljes szószerkezet helyességét. Az egyes morfémáknak tulajdonságai, jegyei vannak, és ezek segítségével megszorítások fogalmazhatók meg a szomszédos morfémák között, amelyek leírják, hogy mely elemek kapcsolódhatnak egymással. Elemzés közben az

elemző folyamatosan ellenőrzi, hogy a felismert szóelemek tulajdonságai egymással összeférnek-e, illetve, hogy az adott elemzés a lehetséges morfémaszerkezeteket megadó és a távoli morfémák közötti megszorításokat is leíró grammatikai leírásnak is megfelel-e.

A Humor egyes elemzései morfok sorozatából állnak. Minden egyes morfnak van egy lexikai (szótári) és egy felszíni alakja (amilyen alakban a morféma az adott szóalakban megjelenik), és mindegyikhez tartozik egy kategóriacímke. A lexikai és a felszíni alak egybeeshet. Csak akkor szerepel az elemzésben mindkét alak, ha különböznek.

Az 1. táblázatban látható egy elemzés és az egyes részeinek értelmezése.

1. táblázat Példa egy Humor elemzés értelmezésére

elemzés	
el[IK]+megy[IGE]=men+tek[t2]	
morf	**jelentése**
el[IK]	*el* morféma, igekötő (IK)
megy[IGE]=men	*megy* lexikai alak itt eltérő, *men* felszíni alakkal, ige
tek[t2]	jelen idő többes szám 2. személy kijelentő mód

Az elemzési alternatívák kezelése

Az elemző gyakran több elemzést ad ugyanahhoz a szóhoz. Nemcsak a többértelmű szavak esetében fordul ez elő, hanem gyakran ugyanazon lexéma paradigmájának különböző tagjai is egybeesnek (pl. *keresnék* 'ők azt', ill. 'én valamit'). A program gyakran váratlan elemzésekkel is megörvendeztet, l. pl. a *fejetlenség* szó utolsó elemzését a 2. táblázatban.

2. táblázat: Példák a Humor többértelmű elemzéseire

szó	**elemzés**	**értelmezés**
várnak	vár[FN]+nak[DAT]	*vár* mint főnév, 'annak'
	vár[IGE]+nak[t3]	*vár* mint ige, 'ők …'
alma	alma[FN]+[NOM]	*alma*
	alom[FN]=alm+a[PSe3]+[NOM]	*alom* (az állat *alma*)
fejetlenség	fejetlenség[FN]+[NOM]	'káosz'
	fejetlen[MN]+ség[_PROP]+[NOM]	'kaotikus mivolt'
	fej[FN]+etlen[_FFOSZT]+ség[_PROP]+[NOM]	'hogy nincs feje'
	fej[IGE]+etlen[_IFOSZT]+ség[_PROP]+[NOM]	'hogy nincs megfejve'
szerelem	szerelem[FN]+[NOM]	*szerelem* főnév
	szerel[IGE]+em[Te1]	*szerel* ige (*szerelem* a biciklimet)
	szer[FN]+elem[FN]+[NOM]	*szer + elem* összetett szó

Humor elemzőre épülő tövesítés algoritmusa

Az általunk kifejlesztett lemmatizáló modul a Humor elemzéseire épül: az elemzésben szereplő morfokból azok címkéje által meghatározott szerepük figyelembevételével építi fel a szó tövét.

A tő építésekor az abban szereplő morfémák felszíni alakját használjuk fel, kivéve az utolsó tőalkotó morfémát: ennek a szótári alakja szerepel a tőben. Természetesen kérdés, hogy mely morfémák számítanak tőalkotónak, és melyek nem. A képzők esetén különösen így van. Például az *adósság* szó esetén a *-ság* képzőt levágva *adós* lesz a tő. De ha az *-s* illetve az *-ó* képzőt is levágjuk, akkor már *adó* illetve *ad* lesz a szó töve. Látható, hogy meghatározó a végeredmény tő szempontjából, hogy egy adott képzőt tőalkotónak tekintünk-e vagy sem. Általában érdemes figyelembe venni, hogy az adott alkalmazás szempontjából mi lehet a legkedvezőbb megoldás. Minél több képző levágása növelheti a tövesítés fedését (kevesebb potenciálisan releváns találatot veszítünk el), de ronthatja a pontosságát (több nem releváns találat áll elő). Néhány példa látható a 3. táblázatban a Humor elemzéseire és a belőlük előállított tövekre.

3. táblázat: Humor elemzések és a belőlük kiszámolt tövek, adott képzőbeállítások mellett

input: szó	**elemzés**	**output: a szó töve**
várnak	vár[FN] +nak[DAT]	vár[FN]
	vár[IGE]+nak[t3]	vár[IGE]
alma	alma[FN]+[NOM]	alma[FN]
	alom[FN]=alm+a[PSe3]+[NOM]	alom[FN]
fejetlenség	fejetlenség[FN]+[NOM]	fejetlenség[FN]
	fejetlen[MN]+ség[_PROP]+[NOM]	fejetlenség[FN]
	fej[FN]+etlen[_FFOSZT]+ség[_PROP]+[NOM]	fejetlenség[FN]
	fej[IGE]+etlen[_IFOSZT]+ség[_PROP]+[NOM]	fejetlenség[FN]
adósság	adósság[FN]+[NOM]	adósság[FN]
	adós[FN]+ság[_COL]+[NOM]	adósság[FN]
	adós[MN]+ság[_PROP]+[NOM]	adósság[FN]
	adó[FN]+s[_SKEP]+ság[_PROP]+[NOM]	adósság[FN]
	ad[IGE]+ós[_SZOK]+ság[_PROP]+[NOM]	adósság[FN]

Hogy a morféma tőalkotónak számít-e, a morféma címkéje dönti el. Az adott morfológiai lexikonban használt címkék halmaza és hogy pontosan mely képzőket milyen esetben érdemes tőalkotónak tekinteni, illetve milyen egyéb beállításokra van szükség, a nyelvtől és az adott alkalmazástól is függ. A tövesítés szempontjából releváns címkekészletek leírására ezért a lemmatizáló egy külön konfigurációs fájlt használ. Így a tövesítő modulba nem kell nyelvspecifikus adatokat bedrótozni, másrészt a korábban bemutatott képzők problémáját ilyen módon hangolni lehet az adott

feladathoz. Ez nagy szabadságot és hangolhatóságot biztosít a modulnak.

Az általunk definiált konfigurációsfájl-formátum felülről kompatibilis a Tihanyi László (MorphoLogic) által korábban C nyelven implementált és szintén a Humor elemzőre épülő HelyesLem lemmatizáló hasonló konfigurációs fájljának formátumával. A HelyesLem főként egyszálú használatra készült, és bár az alapvető lemmatizálási feladat megoldására alkalmas, számos bonyolultabb szókonstrukció esetében nem helyes lemmát ad vissza. Az ebben a cikkben bemutatott implementáció C++ alapú, többszálú (multithread safe), és a lemmatizálási algoritmust, illetve a morfémák osztályozását kiegészítettük mindazokkal a finomságokkal, amelyek a HelyesLem által nem helyesen kezelt szerkezetek esetében is helyes eredményt adnak.

A tövesítő hangolható paraméterei

Bár a lemmatizáló algoritmusának és konfigurációs lehetőségeinek fejlesztésekor a magyar alaktani szerkezetek teljes és helyes lefedése volt az elsődleges célunk, az eszköz általános, és nyelvfüggetlenül használható. A Humor morfológiai elemzőhöz számos nyelvre készült morfológiai leírás, ezek mindegyikére jól használható az itt ismertetett lemmatizáló is, beleértve például azokat az újlatin nyelveket is, amelyeknél – hasonlóan a magyar ikerszavakhoz – a szóalakok belsejében is előfordulnak ragok.

A lemmatizáló konfigurációs fájljában (amelynek formátuma a Windows ini fájljainak formátumára hasonlít) különböző szekciók szolgálnak az egyes morféma-osztályokba tartozó morfémák címkéinek megadására, illetve különböző konverziók és speciális szűrők definiálására. A konfigurációs fájlban található paraméterezhető tulajdonságok egy-egy nyelvi jelenség kezelésére születtek, majd többször finomítottuk a leírásokat és a lemmatizáló algoritmusát is. A jelenlegi változat képes az összes eddig felmerült eset kezelésére.

A két legalapvetőbb szekció a tőalkotó morfémák címkéinek *(stem)* és a képzők eredő szófajának *(conversion)* megadására szolgál (pl. hogy az -i képző melléknevet hoz létre). Emellett lehetőség van olyan címkekonverziók megadására is, amely nem jár azzal, hogy képzőnek is tekintse az algoritmus az adott morfémát *(tag replace).* Ez lehetőséget ad többek között a morfológiai elemző által visszaadott címkekészlet egyszerűsítésére vagy egyszerűen a címkék más alkalmazás által várt formára való hozására.

A tő szófaját az utolsó tőalkotó morféma szófaja határozza meg. Ha ez képző, akkor a fent definiált módon a képzett szófaj szerepel az eredményben. Ez alól kivételt jelent például az elliptikus szerkezetekben szereplő szóösszetételi tagok végén álló a hibás központozásból adódó

szóvégi kötőjel vagy gondolatjel/hosszú kötőjel. Ezek esetében az eredő szófaj az írásjel szófaja lenne (mert ezek az elemek tőalkotók, amikor valóban a szó belsejében állnak), azonban szó belseji írásjelként *(internal punctuation)* való megadásukkal ez elkerülhető. Az egyébként is csak a szó szélein megjelenő írásjelek nem okoznak hasonló problémát, ezeket a program egyszerűen levágja.

A tő kezdetét megelőző címkék alapesetben nem kerülnek be az eredménybe. A prefixumként *(prefix)* megadott morfémák ez alól kivételt képeznek. A magyarban ilyen például a felsőfok jele. A tövet követő címkéket inflexiónak tekinti az algoritmus, és – amennyiben az eszközt használó alkalmazásnak (például egy morfológiai egyértelműsítő programnak) szüksége van erre az információra – a szófajcímkét követően visszaadja.

A tövesítő tudja jelezni a szóösszetételi határ(oka)t. Ehhez két konfigurációs szekció beállítása szükséges: az egyik az összetételi tagként szereplő morfémák címkéinek megadására szolgál *(compound member),* a másik azon morfémák megadására, amelyek egyikének mindenképpen szerepelniük kell egy összetételben *(compound must have).*

Egy kötőjeles szó akkor lehet összetett szó (pl. *Árpád-ház*), ha a kötőjel előtt olyan címke áll, amelyet a *compound before hyphen* szekcióban megadunk. Erre azért van szükség, mert a magyar Humor elemző elemzéseiben ezen a helyen [NOM] (alanyeset) címke is állhat, amely önmagában nem tőalkotó elem, ezen kívül az ikerszavak kezelésének feltétele, hogy ezeket meg tudjuk különböztetni a sima összetételektől.

Korábban említettük, hogy az alkalmazástól függhet, hogy bizonyos képzőket tőalkotónak érdemes-e tekinteni, vagy sem. Sok esetben érdemes például az igenévképzőket nem tőalkotónak tekinteni, így a *mosó, mosott, mosandó* alakokat, és ezek továbbragozott alakjait a *mos* tőre vezethetjük vissza. Hasonlóképpen a melléknevek esetében a fokozást a legtöbb feladatban érdemes inflexiónak tekinteni, és a közép- vagy felsőfokú alakokat az alapalakra visszavezetni. Azokban az esetekben azonban, amikor egy ilyen elemet olyan másik képző követ, amelyet tőalkotónak akarunk tekinteni (pl. *megnagyobbít),* az egyébként kváziinflexióként kezelt elemet is a tő részének kell tekinteni. Ennek az a módja, hogy az adott morfémát felsoroljuk a képzők között, de nem szerepeltetjük a tőalkotó morfémák között. Ezek feltételesen tőalkotóvá válnak abban az esetben, ha tőalkotó morféma (képző vagy tő) követi őket.

Az *-ó, -ás* vagy *-s* képző esetleges kváziinflexióként való kezelése hibás eredményhez vezetne az olyan szerkezetekben, mint a *kőtörő*, *nagybefektető* vagy *háromemeletes,* hiszen ezek lemmájaként hibásan a *kőtör, nagybefektet,*

háromemelet alakok állnának elő. Itt nem követi más tőalkotó elem ezeket a morfémákat, mégsem hagyhatók ki a tőből. Ezt a hibát úgy küszöböltük ki, hogy ezeket a képzőket összetételekben tőalkotónak tekintendő elemekként definiáljuk *(stem if compound),* és az algoritmust is ennek megfelelően módosítottuk.

A tő meghatározásának akár az algoritmusa is paraméterezhető: a *reg* szekcióban megadható, hogy az egyes címkékre illeszkedő morfémáknak a felszíni vagy a szótári alakját számítsa bele a tőbe. Például a `((?:(?:FN|NOM|KJ) )+)(FN|KJ) => {surf}\1 {lex}\2` bejegyzés jelentése: a bal oldali reguláris kifejezés ha illeszkedik, akkor a nyíl jobb oldalán megadott módon az első zárójelezett csoportnak a felszíni alakja kerül a tőbe *(surf)*, a második csoport pedig a lexikális alakjával szerepel. A magyarra implementált alapalgoritmus eddig minden általunk kezelt nyelvre alkalmazhatónak bizonyult. A reguláris kifejezéseken alapuló kiegészítés olyan szerkezetek esetében használható, amelyekre a magyarra kifejlesztett algoritmus esetleg nem ad kielégítő eredményt.

Végezetül lehetőséget biztosít a tövesítő arra, hogy a morfológiai elemző bizonyos elemzéseit egyszerűen kihagyja a *pattern to delete* szekcióban megadott reguláris kifejezésekre illeszkedő elemzések kiszűrésével. Ily módon lehet megszabadulni az esetleges téves összetételektől (pl. *anyó+som, szak+adás)* vagy az adott alkalmazásban nem kívánatos elemzésektől (pl. légy→van, román→roma). Az alábbi minta például a téves *adás/adó* végű összetételek kiszűrésére szolgál:

```
(ár|borz|fog|hal|láz|mar|rag|szak|tag)\[FN\]\+ad(\[IGE|(ó|ás)\[FN)\]
```

Az ikerszavak kezelése

Az ikerszavak *(jövök-megyek, ágával-bogával, okosat-jót)* kezelése különös körültekintést igényel. Névszók és igék is alkothatnak ikerszavakat, és ezekben a két tőnek azonos szófajúnak kell lennie, és azonos toldalékokat kell viselniük (amelyek adott esetben képzők is lehetnek). Maga az elemző nem ellenőrzi a toldaléksorozatok azonosságát, és a szófaji megszorításokat sem teljes körűen. Ezeket az ellenőrzéseket a lemmatizáló szintjén implementáltuk. Az ilyen elemzések csak akkor jelennek meg a lemmatizáló kimenetén, ha helyes elemzés nincs, és ilyenkor is hibás szóként jelöli meg őket.

```
>kastélynak-várnak
kastély-vár[FN][DAT]
>kastélyt-várnak
kastély-vár<incorrect word>[FN][DAT]
kastély-vár<incorrect word>[IGE][t3]
```

Cache

A morfológiai elemzés kiszámítása időbe telik. Nagyobb szöveg indexelésénél jelentős gyorsulás érhető el, ha cache-t használunk, azaz ha az elemzéseket és a lemmatizáló által előállított töveket csak egyszer számoljuk ki, később a gyorsítótárból vesszük elő. Így a lemmatizáló/indexelő alkalmazás futásimemória-igényének bizonyos fokú növekedése árán jelentős, akár 8–10-szeres gyorsulást érhetünk el. A memóriaigény/elért gyorsítás arány optimalizálása érdekében a cache-nek két üzemmódja van. Az egyik üzemmód a cache-építési szakasz, ekkor a bemeneti szavakhoz letárolja az eredményül kapott töveket, és kilépéskor ezeket fájlba menti. (Természetesen ilyenkor nemcsak építi a cache-t, hanem használja is.). Mivel minden szövegkorpuszban a különböző szóalakok nagyobb része csak egyszer fordul elő, sok szöveg tövesítésekor nem érdemes minden szóalakot eltárolni a cache-ben, mert ez komoly memóriaigény-növekedést jelenthet. Ehelyett nagyméretű korpuszból szóalak-gyakorisági listát készítve csak a gyakori szóalakokat érdemes a cache-ben eltárolni, így tudjuk az alkalmazás sebességét korlátozott memória-többletráfordítással a leghatékonyabban növelni. A cache másik, használati üzemmódjában a lemmatizáló a fájlba mentett cache-t csak olvasható módon nyitja meg, és használat közben nem ad hozzá újabb szavakat.

Kivételszótár és ragozó sajátszótár

Előfordul, hogy egy-egy szó ismeretlen a Humor elemző számára. Folyamatosan új szavak kerülnek a nyelvbe, a nevek halmaza sem felsorolható, a legnagyobb körültekintés ellenére is a morfológiai elemző számára ismeretlen szó normálisnak tekinthető. Erre fel kell készülnie a modult használó alkalmazásnak is.

Ha egy szó ismeretlen a morfológiai elemző számára, akkor természetesen a tövesítő számára is az. Új szavakat a morfológiai lexikon újrakompilálásával lehet felvenni, ez időigényes és szakértelmet igénylő feladat. Emiatt felmerült az igény, hogy a tövesítő modul támogassa az új szavak felvételét. Erre kétféle megoldást implementáltunk. Az egyik egy egyszerű sajátszótár, amiben az ott felsorolt adott input szóalakokhoz megadhatjuk a hozzájuk tartozó töv(ek)et. Ez egyben másik lehetőséget ad a korábban említett reguláriskifejezés-alapú megoldás mellett a nem kívánt tövek kiszűrésére is. Ha egy szóalakhoz a sajátszótárban csak az adott alkalmazásban elvárt töveket adjuk meg, akkor a rendszer ezektől különböző tövet nem ad vissza. Pl. a *román* szóalak töve nem lesz *roma*, csak *román*, az *iránt* szóé pedig nem lesz *Irán* (nem nagybetűérzékeny tövesítés esetén), ha a

kivételszótárban csak ezeket a töveket adjuk meg. Emellett a cache fájl szerkesztése és betöltése is lehetőséget ad az elemzések szűrésére, illetve bővítésére.

A másik megoldás egy ragozó sajátszótár, ahol az új szavak alaktani viselkedését egy a morfológiai elemző által már ismert másik azonos módon ragozott szó megadásával lehet a rendszer tudomására hozni. A ragozó sajátszótár formátuma az alábbi:

<új szó> <ismert, hasonló végződésű és ragozású szó> <kívánt szófaj, opcionális>

például:

ódalog andalog
ódalg andalg
Intel lepel FN
vmi valami
fészbuk mameluk

Elemzés előtt a modul az ismeretlen szó helyére az itt kiválasztott ismert szót helyettesíti be (összetételekben is), így hívja meg a morfológiai elemzőt, majd visszahelyettesíti az eredeti szót az elemzésbe. Így a tövesítő algoritmus már jó elemzést kap, amiből ki tudja számolni a tövet. A szavak felvételére és a felvett szavak ellenőrzéséhez egy a mintaként szolgáló szó megtalálását automatizáló eszköz elkészítését tervezzük.

A lemmatizáló kimenetének beállításai

A lemmatizáló kimenetének részletessége sokféleképpen beállítható. Állítható, hogy csak a tövet adja-e vagy a szófajt is. Jelezze-e a szóösszetételi határokat, visszaadja-e az összes kategóriacímkét, illetve az eredeti elemzést is. A következő beállítások bármelyike egymástól függetlenül bekapcsolható:

- a kimenet csak a töveket tartalmazza (*"alma,alom"*)
- a kimenetben a tő+szófaj szerepel (*"alma[FN],alom[FN]"*),
- a szóösszetételi határokat jelzi a kimeneten (*"ablak+kilincs"*),
- minden morfológiai kategóriát tartalmaz a kimenet, (*képviselőházban => képviselőház[FN][INE]*)
- a kimenet tartalmazza az eredeti Humor elemzéseket is
- a bemenet kis/nagybetű állapotát másolja a kimenetre is

A tövesítő szűrői

A tövesítő bizonyos szavakra több tövet is visszaadhat. Az elemzésekben

akár ugyanaz a tő is ismétlődhet, azonban általában nincs szükség arra, hogy azonos eredmények ismétlődjenek a kimeneten. Emellett egyéb szűrésekre is szükség lehet az adott alkalmazás igényeinek megfelelően:

- ugyanaz a tő csak egyszer szerepeljen
- ugyanaz a tő+szófaj csak egyszer szerepeljen
- az elemző által produktívan előállított összetett szavakat nem adja vissza, ha van más tő is (pl. a *szer+elem* tövet kihagyja, mert a *szerelem* tövet egyben megtalálta
- képzők levágásával nyert töveket ne adja vissza, ha talált a lexikonban más tövet (az *adós* töve nem lesz *ad*, ha az *adós* egyben is szerepelt a lexikonban)
- azon töveket ne adja vissza, amelyek az inputtal egyeznek. Erre tipikusan keresési/indexelési feladatnál lehet szükség, ha az indexelő az eredeti szóalakot eleve automatikusan felveszi az indexbe.

Kiértékelés

A fent bemutatott lemmatizáló teljesítményét néhány szabadon hozzáférhető tövesítőével összevetve a legnagyobb elérhető tövekkel is annotált kézzel ellenőrzött korpuszon, a Szeged Korpusz 2.0-s változatán (Csendes et al. 2005) értékeltük ki. Ebben a korpuszban nagyjából 80 000 mondat szerepel, többféle forrásból (szépirodalom, nyolcadikos és tizedikes tanulók fogalmazásai, újságcikkek többféle napi-, illetve hetilapból, számítástechnikai, jogi és üzleti szövegek). A korpuszban jelölve van minden szó adott kontextusban érvényes töve (lemmája), illetve az egyéb lehetséges lemmák. Sajnos nem minden tő helyes a korpuszban (illetve mint később visszatérünk rá, a korpuszban a lemmatizálás elvei több ponton különböznek attól, ahogy az alább kiértékelt eszközök működnek), de mivel ez a legnagyobb elérhető annotált korpusz magyar nyelvre, és számos nyelvtechnológiai mérés használja, ezt választottuk a mérés alapjául. A kiértékelést számos különböző metrika szerint elvégeztük. Megmértük azoknak a szavaknak az arányát, amelyre egy-egy tövesítő nem adott eredményt (ismeretlen (OOV=out-of-vocabulary) szavak), illetve az egyes tövesítő modulok sebességét.

A különböző lemmatizáló modulok különböző morfológiai címkekészleteket használnak (KR, Humor kód stb.), ezek mindegyike különbözik a Szeged Korpuszban használt MSD kódoktól is, ezért a kiértékelésnél csak a tő helyességét vizsgáltuk, a szófaj- és egyéb morfológiai címkékét nem.

Az egyik vizsgált szabadon hozzáférhető tövesítő a **Hunspell**, amelyet széles körben használnak, főként nyílt forráskódú projektekben. (Jelen sorok írásakor több mint 30 alkalmazás használja, köztük a LibreOffice, OpenOffice, Firefox, Thunderbird, Google Chrome.) Alapvetően helyesírás-ellenőrzésre használják, de tövesíteni is tud. Az implementáció nyelve a C++. Számos nyelvre készült hozzá lexikon.

A következő, részben a Hunspell és az ahhoz készített magyar morfológiai leírás fejlesztésekor szerzett tapasztalatok felhasználásával készült eszköz az OCaml nyelven implementált **Hunmorph (Ocamorph)** (Trón et al. 2005), amely a morphdb.hu adatbázisra épül (Trón et al. 2006). A lexikonfájlok a Hunspellhez hasonló formátumúak (aff/dic), azonban nem teljesen kompatibilisek. A Hunmorph nemcsak tövesíteni tud, hanem teljes morfológiai elemzést ad. Az Ocamorph elemzőn alapul az **Ocastem** lemmatizálómodul, amely kifejezetten információ-visszakereső alkalmazások igényeinek kiszolgálására készült. Az Ocastem alkalmazás egyik legnagyobb előnye, hogy pusztán a lehetséges toldalékok levágásával olyan szavak lemmatizálására is képes, amelyek a szótáraul szolgáló morphdb.hu adatbázisban nem szerepelnek, így minden szóra ad vissza eredményt. Az Ocastem alapbeállítása az, hogy a produktívan összetett szavakat tagjaira bontja, és ezeket külön tőként adja vissza. A Szeged Korpuszban szereplő lemmák nem így vannak megadva, és ez a tövesítő mért eredményeit hátrányosan befolyásolná, ezért az Ocastemet olyan beállítással futtattuk, amely ezt az összetettszó-daraboló működést kikapcsolja. A számos egyéb beállítási lehetőséget is kipróbálva alább a kiértékelésnél a legjobb eredményeket adó fent említett, mindig pontosan egy tövet visszaadó beállítást használtuk.

A magyar **Snowball** tövesítőnek (Tordai & De Rijke 2006) az NLTK fejlesztői csomagban (Bird 2006) szereplő változatát használtuk. Mivel ez a végződéseket egy szótárt nem használó algoritmus alapján vágja le az angolra készült Porter Stemmer (Porter 1980) mintájára, sokszor nem igazi szótövet ad, hanem egy szócsonkot. Előnye viszont, hogy ebből fakadóan az Ocastemhez hasonlóan számára nincs ismeretlen szó, mint az a 4. táblázatban látható.

A **Hunmorph-foma** egy véges állapotú fordítóautomatákon alapuló eszköz, amely a Foma morfológiai elemzőhöz (Hulden 2009) készített, a Hunmorph adatbázisából konvertált magyar morfológiai leíráson alapul. A véges állapotú elemzőimplementáció rendkívül gyors: bár az automata bejárása során a többértelműségek miatt nem elkerülhető hogy az elemző visszalépjen, és újabb bejárási útvonalakat is kipróbáljon, tehát a bejárás nem determinisztikus, mégis ez a tövesítő a leggyorsabb.

Eredmények

A tövesítőmodulok eredményeit számos metrika alapján kiértékeltük. A valódi lemmatizáló modulok által adott elemzések esetében, amelyek a tő mellett morfoszintaktikai annotációt is tartalmaznak, elvileg lehetőség lett volna a szó környezetét figyelembe vevő statisztikai egyértelműsítő (pl. Halácsy et al. 2006, Orosz és Novák 2014) alkalmazására is az adott kontextusban helyes tő kiválasztására. Ettől a méréstől itt eltekintettünk, egyrészt mert szükség lett volna hozzá a Szeged Korpuszban használt MSD kódrendszer és az összes többi eszköz egyedi címkézési rendszere közötti konverzióra, másrészt mert az információ-visszakereső rendszerekben ilyen egyértelműsítő eszközt általában nem használnak. Ehelyett az indexelőrendszerrel és magukkal a tövesítőeszközökkel kapcsolatos különböző feltételezésekkel élve az alábbi méréseket végeztük el.

Hogy az egyes modulok a korpuszban szereplő szóalakok mekkora részére nem adnak vissza tövet, vagyis mekkora az ismeretlen (OOV) szavak aránya, a 4. táblázat tartalmazza.

4. táblázat A tövesítő modulok számára ismeretlen szavak aránya a Szeged Korpuszon

alkorpusz	méret	***Hunspell***	***Hunmorph -foma***	***Hunmorph compound***	***Hunmorph***	***Ocastem***	***Snowball***	***Humor***
szépirodalom	185 436	3,5	2,2	**1,1**	2,4	**0**	**0**	1,5
tanulók	278 497	1,4	1,0	0,7	1,4	**0**	**0**	**0,4**
újságcikk	182 172	4,8	3,8	2,6	5,5	**0**	**0**	**1,1**
IT	175 991	8,2	5,5	4,1	7,9	**0**	**0**	**2,7**
jogi	220 069	7,0	6,8	5,7	7,5	**0**	**0**	**2,0**
üzleti	186 030	8,0	7,8	5,7	9,0	**0**	**0**	**1,4**
összesen	1 228 195	5,4	5,2	3,7	5,7	**0**	**0**	**1,5**

Az algoritmikus Snowball és az ismeretlenszó-elemzést alkalmazó Ocastem minden szóra ad vissza tövet. A szigorúan szótáralapú eszközök közül az itt ismertetett Humor-alapú lemmatizáló lexikona adta a legjobb lefedést a korpuszon (egy részkorpusz kivételével). A Hunmorph elemző a produktív szóösszetétel bekapcsolásával (Hunmorph compound oszlop) közelítette meg ezt leginkább. Bár a Hunspell helyesírás-ellenőrzőként képes produktív összetételek létrehozására, a tövesítetlenül maradt szavak átnézésekor azt tapasztaltuk, hogy valamilyen implementációs hibából kifolyólag ez a tövesítő üzemmódban nem működik jól. Ennél a kiértékelésnél azt is észrevettük, hogy a Hunmorph-foma elemző az igekötős

igékre, összetett számnevekre, illetve az önálló szóként nem előforduló összetételekre olyan elemzést ad vissza, amelyben az első összetételi tag (pl. az igekötő) elvész. Ezt kijavítottuk az elemző forráslexikonának módosításával, és az alább leírt méréseket már ezen a javított lexikonon végeztük.

Következő mérésünknél (5. táblázat) azzal az alapfeltételezéssel éltünk, hogy az indexelőrendszer egyetlen optimális elemzést, illetve tövet vár a tövesítőalkalmazástól, amelynek ezt a kontextus ismerete nélkül kell meghatároznia. A Snowball és az Ocastem esetében ez a feltétel eleve teljesül, a többi eszköz esetén egyenként kimértük, hogy a teljes korpuszon a tövesítő által visszaadott első tő vagy pedig a leghosszabb tő használata adta-e a jobb eredményt (l. 6. táblázat), és azt használtuk a kiértékelésnél. Két részkorpusz kivételével megint a Humor-alapú lemmatizáló adta a legjobb eredményt. A Hunmorph-foma esetében az eredeti nem javított adatbázissal csak 71,5% pontosságot kaptunk.

5. táblázat Tövesítő modulok első/leghosszabb javaslatának pontossága a Szeged Korpuszon

alkorpusz	***stemmer nélkül***	***Hunspell*** *(első tő)*	***Hunmorph-foma*** *(leghosszabb tő)*	***Hunmorph-compound*** *(első tő)*	***Hunmorph*** *(első tő)*	***Ocastem***	***Snowball***	***Humor*** *(leghosszabb tő)*
szépirodalom	52,4	86,6	76,2	86,6	86,4	**88,7**	58,3	88,4
tanulók	52,9	**88,6**	78,1	88,2	88,1	88,0	57,0	88,3
újságcikk	57,3	84,5	75,5	83,1	81,8	88,6	64,7	**92,8**
IT	57,9	81,9	75,8	81,7	79,3	87,9	68,6	**92,5**
jogi	62,0	81,8	77,4	82,4	80,8	86,7	72,4	**93,8**
üzleti	55,5	78,1	68,9	80,2	78,9	87,6	65,2	**91,4**
összesen	56,2	83,9	75,6	84,0	83,0	87,9	64,0	**91,0**

A következő mérésben, melynek az eredménye a 6. táblázat harmadik oszlopában szerepel, arra voltunk kíváncsiak, hogy egy ideális orákulum használata esetén, amely ki tudná választani a helyes elemzést az összes közül, milyen pontosságot kapnánk. Ez a mérés azt adja meg, hogy az esetek mekkora részében szerepel az elemzések között a Szeged Korpuszban megadott lemma. Látható, hogy bár ebben a mérésben is a Humor-alapú lemmatizáló érte el a legjobb eredményt (96%), a 4%-nyi eltérésre nem ad magyarázatot a mindössze 1,5%-nyi ismeretlen szó. Az eltérések oka az, hogy a lemmatizálás a Szeged Korpuszban részben más elveken nyugodott, mint ahogy akár a mi lemmatizálónk, akár a többi eszköz működik. A legfőbb eltérések a következők:

- A *-hat* toldalékos igék (pl. *futhatott*) lemmája a Szeged Korpusz adott verziójában tartalmazza a *-hat* végződést (*fut* helyett *futhat*)
- A ragozott személyes névmások (pl. *rajtam, velünk*) lemmája a Szeged Korpuszban nem a személyes névmás (*én, mi*), hanem a *rajta, vele* alakok.
- A melléknévi igenevek nem az igére vannak visszavezetve, hanem egyszerűen melléknévként vannak annotálva.

6. táblázat A tövesítő modulok pontossága több tőalternatíva esetén más-más kiválasztási módszer mellett

	Első tő	Leghosszabb tő	optimális tőalternatíva-választással elérhető
Hunspell	**83,9**%	83,2%	87,6%
Hunmorph-foma	73,8%	**75,6**%	91,0%
Hunmorph - compound	**84,0**%	78,4%	90,0%
Hunmorph	**83,0%**	82,2%	87,4%
Ocastem	87,9%	87,9%	91,4%[1]
Snowball	64,0%	64,0%	64,0%
Humor	89,6**%**	**91,0%**	**96,0%**

A következő kiértékelésben azt feltételeztük, hogy az egyes tövesítők minden tőjavaslatát az indexbe helyezzük, és minden egyes a korpuszban megadott lemmától eltérő indexbe került tétel hibás találatot jelent (false positive, FP), illetve plusz hibapontot jelent, ha a korpuszban szereplő lemma nem szerepel a javaslatok között (false negative, FN). A helyes lemmák jelentenek jó találatot (true positive, TP). Így pontosságot (P=TP/(TP+FP)) és fedést (R= TP/(TP+FN)) számolva, és ezekből a pontosságot és a fedést egyforma súllyal figyelembe vevő F-pontszámot kiszámolva kaptuk a 7. táblázatban szereplő eredményeket.

7. táblázat Tövesítő modulok hibás alternatíváit szigorúan lepontozó F-pontszáma Szeged Korpuszon

alkorpusz	***stemmer nélkül***	***Hunspell***	***Hunmorph -foma***	***Hunmorph compound***	***Hunmorph***	***Ocastem***	***Snowball***	***Humor***
szépirodalom	68,8	87,5	62,0	37,5	56,1	**94,0**	73,6	89,8
tanulók	69,2	86,9	60,1	40,1	55,9	**93,6**	72,6	88,5
újságcikk	72,8	88,9	62,4	29,5	53,4	**93,9**	78,6	92,5
IT	73,3	90,3	63,3	30,9	53,3	**93,5**	81,3	92,9
jogi	76,5	90,2	64,3	29,0	48,7	**92,9**	84,0	92,3
üzleti	71,3	86,5	60,6	27,0	51,3	**93,4**	78,9	92,3
összesen	72,0	88,3	62,0	32,3	53,1	**93,5**	78,0	91,1

[1] Az Ocastemet olyan beállítással lefuttatva kaptuk ezt az eredményt, hogy az összes lehetséges tövet adja vissza. A többi táblázatban szereplő beállítással itt is 87,9% állna.

Ebben a mérésben a mindig csak egyetlen tövet visszaadó Ocastem lemmatizáló bizonyult a legjobbnak, bár az alapjául szolgáló rengeteg „vicces” elemzést generáló Hunmorph teljesítménye e szerint a szigorú metrika szerint nagyon messze elmaradt még attól a megoldástól is, ha egyáltalán nem végeztünk tövesítést. A Humor lemmatizáló ebben a megmérettetésben a második legjobb eredményt adta annak ellenére, hogy nem végeztünk szűrést a javaslatain, így az alternatív tövek sok hamis pozitív találatot adtak. Ezt a mérést gyakran úgy végzik el, hogy a mérés az ajánlatok sorrendezését is minősítse. Ilyenkor a pontosságot nem az egész listán számolják, hanem azt mérik meg, hogy a találati listán végigmenve a maximális lefedés elérésekor milyen pontosságot érünk el (precision at maximum recall). Ilyenkor a listákon a helyes tövet követő javaslatokat nem tekintjük téves pozitívnak. Így a 8. táblázatban látható eredményt kapjuk. Ebben a mérésben ismét a Humor a legjobb.

8. táblázat Tövesítő modulok alternatívákat pontozó kiértékelése a Szeged Korpuszon

alkorpusz	méret	***stemmer nélkül***	***Hunspell***	***Hunmorph -foma***	***Hunmorph - compound***	***Hunmorph***	***Ocastem***	***Snowball***	***Humor***
szépirodalom	185 436	68,8	93	89,1	89,6	92,1	94,0	73,6	**94,7**
tanulók	278 497	69,2	94,1	88,4	91,3	93,0	93,6	72,6	**94,1**
újságcikk	182 172	72,8	91,8	90,0	85,9	89,5	93,9	78,6	**95,4**
IT	175 991	73,3	90,3	88,8	86,2	88,1	93,5	81,3	**94,9**
jogi	220 069	76,5	90,2	87,4	87,3	88,7	92,9	84,0	**93,7**
üzleti	186 030	71,3	88,2	89,1	83,3	87,7	93,4	78,9	**95,9**
összesen	1 228 195	72,0	91,5	88,6	87,6	90,1	93,5	78,0	**94,7**

Kiértékeltük a tövesítőket abból a szempontból is, hogy a korpuszban szereplő szóalakoknak a korpuszban ténylegesen szereplő lemmáit milyen jól fedik le. Minden szóalakhoz felvettük a korpuszannotációban szereplő összes lemmát, és ezt a halmazt hasonlítottuk össze az egyes tövesítők által visszaadott tövek halmazával. A metszet TP, a csak a korpuszban szereplő lemmák FN, a csak a tövesítő által javasolt tövek pedig FP jelölést kaptak. Így kaptuk a 9. táblázatban látható eredményeket. Ebben a kiértékelésben ismét a Humor-alapú lemmatizáló végzett az élen.

9. táblázat F-pontszám az összes lehetséges helyes tő figyelembevételével a Szeged Korpuszon

alkorpusz	***stemmer nélkül***	***Hunspell***	***Hunmorph -foma***	***Hunmorph compound***	***Hunmorph***	***Ocastem***	***Snowball***	***Humor***
szépirodalom	57,6	88,2	80,7	63,3	74,1	83,1	57,9	**91,0**
tanulók	60,6	86,4	79,5	64,5	74,1	82,1	56,1	**88,2**
újságcikk	61,1	86,2	82,7	58,6	70,7	84,1	64,4	**93,1**
IT	62,0	84,4	82,4	58,9	68,4	83,1	68,3	**91,7**
jogi	64,5	84,1	83,0	59,0	67,6	83,1	72,4	**91,2**
üzleti	61,3	82,3	81,0	53,6	65,9	82,0	64,4	**93,2**
összesen	61,2	85,4	81,4	59,9	70,3	82,9	63,4	**91,1**

Természetesen egy alkalmazásnál fontos a tövesítés sebessége, egy újraindexelés futási idejét ez nagyban meghatározza. A 10. táblázatban láthatók a Szeged Korpusz elemzésekor mért futási idők, illetve az egyes modulok sebessége. Itt a cache bekapcsolásával 3-szoros gyorsulást értünk el a Humor-alapú lemmatizáló esetében.

10. táblázat A tövesítőmodulok sebessége

	Futási idő (1.2M token)	**token/s**
Hunspell	847s	1 450
Hunmorph-foma	**44,1s**	27 850
Snowball	70,2s	17 495
Humor	163,2s (cache-sel: 59,9s)	7 525 (20 503)
Hunmorph	74m24,8s	275
Ocastem	21m25s	955

Az algoritmus alkalmazása más feladatokra

A lemmatizálóhoz sok szempontból igen hasonló algoritmusok alkalmazásával oldható meg a morfológiai elemző kimenetének feldolgozásával számos más szövegfeldolgozási feladat, például az ékezet nélkül írt szövegekből az ékezetes szöveg automatikus helyreállítása. Alább bemutatjuk, hogy ilyen jellegű feladatok megoldására hogyan adaptáltuk a lemmatizálót.

Ékezetesítés

A mobileszközök elterjedésével az utóbbi időben ismét megnőtt az ékezet nélkül írt szövegek aránya, amelyek nyilvánosan elérhető felületeken is megjelennek. Ennek oka egyrészről a beviteli módban keresendő: a felhasználó billentyűzete például nem magyar, vagy olyan mobil eszközt használ, amelyen az ékezetes betűk begépelése nehézkes. Másik tipikus oka a

felhasználói szokás: így egyszerűbb. Ez főleg SMS üzeneteken, chatszobákban (vagy az email hőskorában) szocializálódott felhasználók esetén fordul elő: ott még nem volt ékezet, és ezt a felhasználó megszokta. Időnként még ma is előfordul, hogy kódolási hibás emailt kapunk, amiben olvashatatlanok lettek az ékezetes betűk. Vannak olyan felhasználók, akik – biztos, ami biztos – nem csak fájlnevekben nem használnak ékezeteket, hanem e-mailekben sem.

Az ember számára általában nem jelent problémát az ékezetek nélkül írt szövegek értelmezése, de az automatikus nyelvfeldolgozó eszközök nincsenek felkészülve az ilyen szövegek feldolgozására. Ezért kísérletet tettünk arra, hogy ezen szövegeket automatikus módszerrel, minél jobb minőségben ékezetes szöveggé alakítsuk.

Jelen cikk írásakor nem volt elérhető ékezetesítő megoldás magyar nyelvre. Több cikk is foglalkozik a témával (Tarján et al. 2013; Kornai & Tóth 1997; Zainkó & Németh 2010), egyikben egy online demó linkje is szerepelt, de sajnos nem volt már elérhető.

Ékezetesítés morfológiai elemzés alapján

A feladat megoldásához úgy módosítottuk a morfológiai elemző lexikonát, hogy a felszíni alakok helyére az ékezet nélküli felszíni alakok kerültek, a lexikai alakokat pedig az ékezetes felszíni alakokkal helyettesítettük. Ékezet nélküli szavak elemzésekor így a morfémák lexikai alakjai hozzák az ékezetes alakot (11. táblázat). Ettől a módosítástól természetesen az elemző csak ékezet nélküli szavakat tud majd elemezni, de jelen feladatnál éppen ez a cél. A korábban bemutatott viszonylag bonyolult tövesítési algoritmus helyett egyszerűen a lexikai alakokat kell konkatenálni, és megkapjuk az ékezetes alakot. Ezt az átalakítást mutatja be a 11. táblázat, ahol az ékezet nélkül érkező szavakat megelemzi a (módosított lexikonú) Humor elemző, és az elemzés lexikai alakjai tartalmazzák az ékezetes alakot.

11. táblázat: Példa a Humor normál- és ékezet nélküli elemzésére, ez az ékezetesítés alapötlete

	bemenet	**Humor elemzés**	**kimenet**
elemző lexikonnal	kutyának	kutya[FN]=kutyá+nak[DAT]	kutya[FN][DAT]
	távolításuk	távolít[IGE]+ás[_IF]+uk[PSt3]+[NOM]	távolítás[FN] [PSt3][NOM]
ékezetesítő lexikonnal	kutyanak	kutyá[FN]=kutya+nak[DAT]	kutyának
	tavolitasuk	távolít[IGE]=tavolit+ás[IF]=as+uk[PSt3]	távolításuk

Az ékezetesítés szempontjából az számít többértelműségnek, ha egy adott

(ékezet nélküli) szónak többféle ékezetes alakja lehetséges. A 12. táblázatban látható, hogy az emberek számára is nyilvánvaló többértelműségek mellett olyan alakok is előállhatnak, amit elméletileg valóban le lehet írni, de soha nem (vagy nagyon ritkán) használjuk: *ősszé, villamosmegállóbán, címkéjé,* stb. Ez utóbbiak abból is adódnak, hogy a Humor produktívan megengedi a szóösszetételeket, és így olyan szavakat is összeilleszt, amelyek meglepőek (például címkéje=cím+kéje, amelyhez felkínálja a *címkéjé* alakot). A 12. táblázatban látható ékezetes alternatívák gyakoriságát a webkorpuszból vettük (Halácsy et al. 2004; Kornai et al. 2006). Jól látható, hogy bizonyos alakok nem (vagy nagyon ritkán) fordulnak elő a valóságban.

Az alternatívák kezelésénél annyit szeretnénk elérni, hogy a legvalószínűbb alak legyen az első. A kevésbé jó ötleteket nem akarjuk letiltani, elegendő, ha hátrébb soroljuk őket.

Ennélfogva a nagyon ritka ékezetes alakokat egyszerű szógyakoriság alapján hátrébb sorolhatjuk.

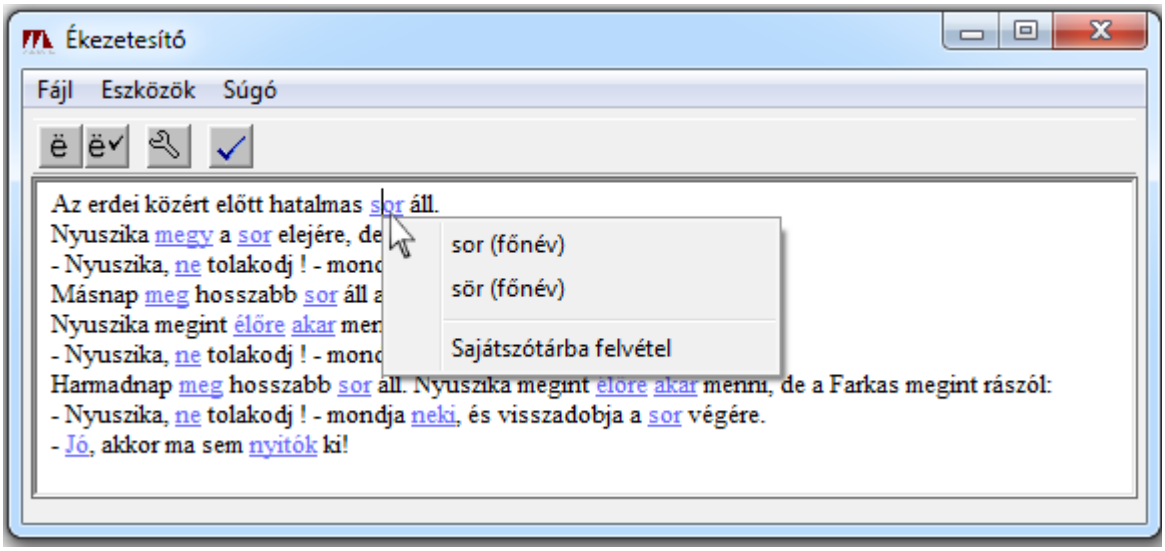
12. táblázat: Többértelműségek az ékezetesítésben, a webkorpuszbeli előfordulásokkal

Ékezet nélküli szó	**Ékezetes alternatívák**	**Előfordulás**
vereb	veréb	743
	véreb	111
fokabel	főkábel	7
	fókabél	3
erintettel	érintettel	264
	érintettél	12
toroljuk	toroljuk	7
	töröljük	1 321
reszletet	részletet	2 722
	részletét	1 757
ugy	úgy	504 105
	ügy	24 404
úr	úr	87 525
	űr	1 466
ossze	össze	124 872
	ősszé	5
testuket	testüket	1 022
	testükét	0
	testűket	0
	testűkét	0
cimkeje	címkéje	114
	címkéjé (összetett szó: cím+kéjé)	0

Három módszert próbáltunk ki a többértelmű ékezetes alternatívák kezelésére. A legegyszerűbb, amikor az első elemzést használjuk. A második módszer szerint gyakoriságalapú döntést hozunk. Pusztán a szóalakok

korpuszbeli gyakorisága alapján jól kiszűrhetőek a téves ékezetes alakok: pl a *villamosmegállóbán*-típusú álösszetételek, ami csak a gép számára alternatíva, emberek nem használják. Nem támaszkodhattunk azonban kizárólag a szóalakok korpuszbeli gyakoriságára, hiszen – többek között a nyelv ragozó mivolta miatt – teljesen értelmes szóalakok sem szerepelnek még igen nagy korpuszokban sem. Ezért a lemmák és a toldaléksorozatok gyakoriságát is figyelembe vettük a rangsorolásnál – a felszíni szóalakénál kisebb súllyal (lényegében csak abban az esetben, ha a szóalak nem szerepelt a korpuszban).

Végezetül egy betűhármasokon alapuló alternatívaválasztási módszert is kipróbáltunk: tanításkor az egyes ékezetes szavakat (illetve ha egyértelmű, akkor tövüket) a szomszédos szavaik trigramjaival (betűhármasaival) együtt tároltuk le. Használatkor pedig ugyanígy a szomszédos szavak trigramjaival kerestük meg a legtöbb trigrammal egyezőt.



1. ábra: Az ékezetesítő alkalmazás egy többértelmű szónál

Az ékezetesítés kiértékelésénél az alap (baseline) az ékezet nélküli szöveg volt, azaz amikor a program nem csinál semmit. Egy 67 ezer szavas korpuszon értékeltük ki a 2,8 millió szavas korpuszon betanított modelleket (13. táblázat). Legjobban a gyakoriság-alapú modell teljesített, mert a szövegkörnyezetet is figyelembe vevő trigram modell nem számol a gyakoriságokkal.

13. táblázat: Az ékezetesítés többértelműségét kezelő megoldások összehasonlítása

	szópontosság	magánhangzó-pontosság
nincs ékezetesítés	53,0	71,7
első alternatíva	90,2	94,9
szógyakoriság	**94,3**	**97,2**
trigram	92,3	96,0

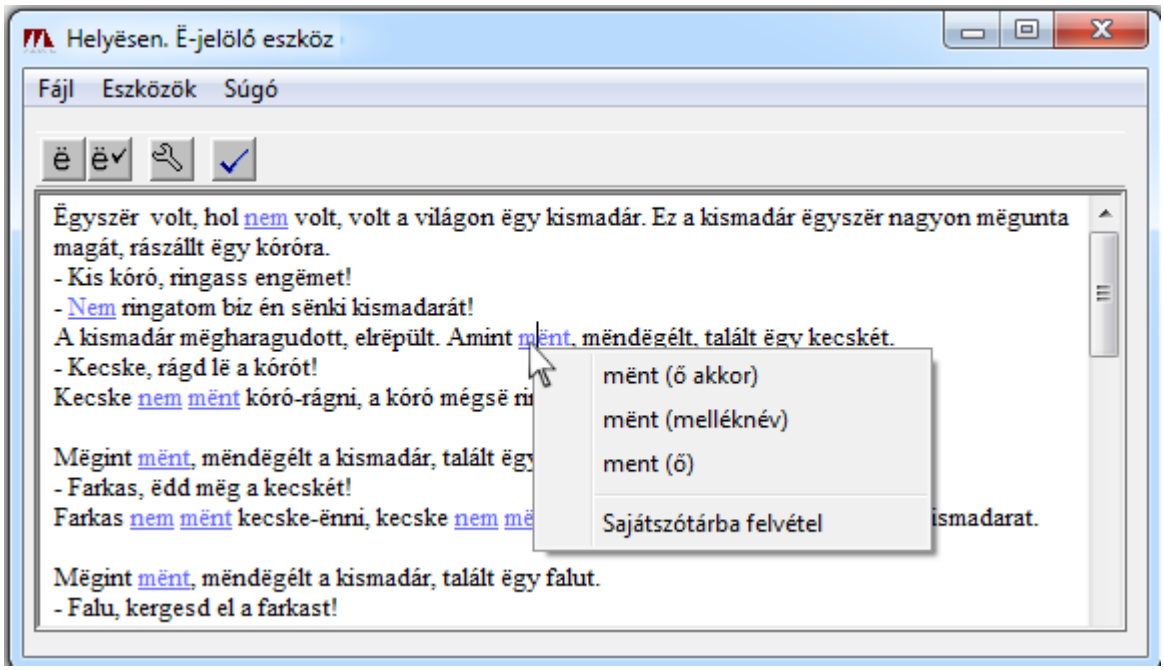
Automatikus *ë*-jelölő alkalmazás

A Bárczi Géza Kiejtési Alapítvány felkérésére készítettünk 2005-ben az ékezetesítőhöz hasonló alkalmazásként egy a félzárt *ë* hang írásbeli jelölését segítő automatikus eszközt (Novák és Endrédy 2005). Az eszközhöz készült lexikonban az *ë* fonémák jelölésével kiegészített felszíni alakok kerültek a lexikai alak helyére, így a tövesítő eszköz ékezetesítő üzemmódban használva éppen a kívánt feladatot végzi el, ahogy az a 14. táblázatban látható. A toldalékmodellt magunk adaptáltuk a feladathoz, a tőtárban az *ë* fonémák jelölését és az elöl képzett nyitótövek azonosítását az alapítvány munkatársai, Buvári Márta és Mészáros András végezték el.

14. táblázat: Zárt ë átalakító algoritmusa azonos az ékezetesítőével

	bemenet	**Elemzés**	**kimenet**
elemző	elmentem	el[IK]+megy[IGE]=men+t[MIB]+ek[PL]+[NOM]	elmentem
zárt ë átalakító	elmentem	el[IK]+mën[IGE]=men+t[MIB]+ek[PL]+[NOM]	elmëntem

A nyelvi modult egy rich text szövegszerkesztőbe integráltuk, ahol az egyes többértelmű szavak aláhúzással jelennek meg, majd jobb egérgomb hatására láthatóak az alternatívák egy pop-up menüben (2. ábra).



2. ábra: A zárt-ë átalakító alkalmazás (ë-jelölő), egy többértelmű szónál

Összefoglalás

Jelen cikkben bemutattunk egy a Humor morfológiai elemzőn alapuló lemmatizáló modult, amelynek a magyar nyelvre alkalmazott változatának teljesítményét összehasonlítottuk más tövesítő modulokkal minőség és sebesség tekintetében a kézzel ellenőrzött annotációt tartalmazó Szeged Korpusz segítségével. A bemutatott lemmatizáló működése egy konfigurációs fájlon keresztül hangolható, az adott feladat és nyelv sajátosságainak megfelelően. Az elkészült tövesítő a legtöbb tekintetben jobb eredményeket ért el, mint a pillanatnyilag elérhető más megoldások, és számos cég, illetve szervezet alkalmazásaiban is felhasználásra került (Microsoft Indexing Service, Országos Atomenergetikai Hivatal, MTI, PolyMeta kereső).

A Humorra épülő tövesítő algoritmust egyéb feladatokra is adaptáltuk. A cikkben bemutatott ékezetesítő eszköz 94,3% szópontossággal (97,2% magánhangzó-pontossággal) képes ékezet nélküli szövegekben az ékezeteket helyreállítani, és a további kézi javításhoz kényelmes felhasználói felületet nyújt. Az elkészült eszközt egy másik hasonló feladatra is adaptáltuk. Ez az alkalmazás a félzárt ë hangzó előfordulásait jelöli automatikusan magyar nyelvű szövegekben.

Irodalom

Bird, Steven. (2006) NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, 69–72. Association for Computational Linguistics.

Csendes D., Csirik J., Gyimóthy T. & Kocsor A. (2005) The Szeged Treebank. In *Lecture Notes in Computer Science: Text, Speech and Dialogue*, 123–31. Springer.

Halácsy P., Kornai A., Németh L., Rung A., Szakadát I. & Trón V. (2004) Creating Open Language Resources for Hungarian. *Proceedings of 4th Conference on Language Resources and Evaluation (LREC)*, 203–10.

Halácsy P., Kornai A., Oravecz Cs., Trón V. & Dániel V. (2006) Using a Morphological Analyzer in High Precision POS Tagging of Hungarian. *Proceedings of 5th Conference on Language Resources and Evaluation (LREC)*, 2245–48.

Hulden, Mans. (2009) Foma: A Finite-State Compiler and Library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, 29–32. Association for Computational Linguistics.

Kornai A., Halácsy P., Nagy V., Oravecz Cs., Trón V. & Varga D. (2006) Web-Based Frequency Dictionaries for Medium Density Languages. In *Proceedings of the 2nd International Workshop on Web as Corpus*, 1–8. Association for Computational Linguistics.

Novák A. (2003) Milyen a jó Humor. In *Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)*, 138–145, Szegedi Tudományegyetem.

Novák A. & Endrédy I. (2005) Automatikus ë-jelölő program. In *A 3. Magyar Számítógépes Nyelvészeti Konferencia Előadásai,* 453–54. Szeged

Orosz Gy., Novák A. (2013) Purepos 2.0: a hybrid tool for morphological disambiguation. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov (Eds.) *Proceedings of the international conference Recent Advances In Natural Language Processing RANLP 2013,* 539–545. Hissar, Bulgaria.

Porter, Martin F. (1980) An Algorithm for Suffix Stripping. *Program* 14 (3): 130–37.

Prószéky G. & Kis B. (1999) A Unification-Based Approach to Morpho-Syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In *ACL*, edited by Robert Dale and Kenneth Ward Church. ACL.

Tordai, Anna, and Maarten De Rijke. (2006) *Four Stemmers and a Funeral: Stemming in Hungarian at Clef 2005*. Springer.

Trón V., Halácsy P., Rebrus P., Rung A., Vajda P. & Simon E. (2006) Morphdb. Hu: Hungarian Lexical Database and Morphological Grammar. In *Proceedings of 5th International Conference on Language Resources and Evaluation*, 1670–73.

Trón V., Kornai A., Gyepesi Gy., Németh L., Halácsy P. & Varga D. (2005) Hunmorph: Open Source Word Analysis. In *Proceedings of the Workshop on Software*, 77–85. Association for Computational Linguistics.