Mozaik nyelvmodell az AnaGramma elemzőhöz

Indig Balázs[1,2], Laki László[1,2], Prószéky Gábor[1,2,3]

[1] MTA–PPKE Magyar Nyelvtechnológiai Kutatócsoport
[2] Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar
[3] MorphoLogic
e-mail:{indig.balazs, laki.laszlo, proszeky.gabor}@itk.ppke.hu

Kivonat Cikkünkben bemutatjuk az elemző rendszerünkhöz a rendelkezésünkre álló nagyméretű magyar nyelvű korpuszok felhasználásával készített modult, amely szimulálni tudja az emberi elemzőkön megfigyelt jelenséget, miszerint bizonyos gyakori szerkezetek feldolgozása egyfajta gyorsítótárazás segítségével az átlagosnál gyorsabb. Létrehoztunk egy olyan rendszert, amellyel 3-nál magasabb gramok esetén, több faktor kombinálásával gyakori mintákat tud előállítani. Megvizsgáltuk a keletkezett mintákat, a szintaktikai elemzés gyorsításának szempontjából, beleértve az őket alkotó példák különböző teljes kifejtésű eloszlásait. Az ilyen minták megfigyelésével a szakértő szemlélő további ötleteket nyerhet, a korpuszon megfigyelhető jelenségek keresésére. Felsorolunk továbbá néhány az elemző szempontjából érdekes példákat is.

Kulcsszavak: nyelvmodell, korpuszminták, nyelvi elemző, big data

1. Bevezetés

Az MTA–PPKE Magyar Nyelvtechnológiai Kutatócsoport a létező megközelítésektől merőben eltérő nyelvelemzőrendszer létrehozását tűzte ki célul. Az általuk létrehozott AnaGramma nevű [1] pszicholingvisztikai indíttatású elemzőrendszer lényege, hogy működésével az emberi nyelvelemzést modellezi.

Ennek megfelelően a következő főbb tulajdonságokkal rendelkezik: (1) Szigorúan balról jobbra, szavanként dolgozza fel a szöveget, így nincs mód az elemzéshez a mondat azon részének a felhasználására, amely még nem került be a rendszer látókörébe. (2) Az elemző több, független, párhuzamosan futó modulból áll, amik kommunikálnak egymással. (3) Performanciaalapú rendszerként minden olyan nyelvi jelenséget megpróbál feldolgozni, ami leírt szövegekben előfordul, viszont a csak elméleti szinten létező szerkezetek elemzése nem tartozik az elsődleges céljai közé.

2. Mozaik n-gramok

Munkánk során a fent vázolt rendszer egy olyan modulját készítettük el, amely a rendszer egésze által támasztott kritériumoknak megfelel. A modul feladata egy olyan adatbázis felépítése és lekérdezése korpuszok felhasználásával, amely

tárolja a nyelvben előforduló *gestaltokat*, azaz olyan gyakori szerkezeti mintákat, melyeket az emberi elemző a teljes elemzés helyett *egészleges feldolgozás* segítségével gyorsan, „egy lépésben" kezel. Az előre eltárolt, megelemzett „mentális reprezentáció" egészben hívódik elő a memóriából a szerkezet valósidejű felépítése helyett.

Ilyen minták lehetnek például az *állandósult szókapcsolatok*, a *többszavas kifejezések*, *főnévi frázisok* vagy bármilyen más gyakori, összefüggő szerkezetek, amelyek a korpusz alapján megfigyelhetők. A következő felsorolásban néhány példamondatot gyűjtöttünk össze:

- **Többszavas kifejezések**: „az ördög ügyvédje"
- **Szólásmondás**: „Itt van a kutya elásva."
- **Udvariassági formula**: „Jó [NAPSZAK][ACC]!"
- **Merev szerkezetek**:
  „Az országgyűlés a javaslatot [SZN|DIGIT][NOM] igennel... elfogadta."
- **Igei szerkezetek**: „lemma:es(ik) szó [DEL]"
- **Névelem**: „Petőfi/[VEZ.NÉV] Sándor/[KER.NÉV] utcai/[KÖZT.TÍPUS] általános/[INT.TÍPUS] iskola/[INTÉZMÉNY]"
- **Név + titulus**:
  „Orbán/[VEZ.NÉV] Viktor/[KER.NÉV] Magyarország/[ORSZÁG|SZERVEZET] miniszterelnöke/[TITULUS]"

Az általunk létrehozott modul legfontosabb tulajdonsága, hogy a nyelvi intuíciónkat „utánozza", miszerint kategoriális helyettesítéseket alkalmazva mintákkal leírhatók a nyelvi jelenségek. Ennek alátámasztására nagy korpuszokon (MNSZ 1-2, lásd 1. táblázat) számszerűsítve vizsgáltuk különböző hosszúságú, összefüggő n-gramok számosságát, mert előzetes tapasztalatainkban úgy találtuk, hogy bizonyos minták esetén egyes szóalakok *kategóriáikkal (faktorokkal)* való helyettesítése olyan számszerű összefüggéseket tár fel, amik a hagyományos n-gramok esetén nem látszódnak. Így olyan gyakori mintákat generáltunk a korpuszokból, amelyekben „szükség szerint" a szóalakok helyettesítve lettek a lemmájukkal, illetve a szófaji címkéjükkel. Az így létrejött heterogén felépítésű n-gramokat nevezzük *mozaik n-gramoknak*[4].

1. táblázat. Különböző korpuszok jellemzői

Neve	Mondatok száma	Tokenek száma	Mondatokban az átlagos tokenszám
Szeged Korpusz 2	70 990	1 194 348	16,824
MNSZ1	18 657 302	264 465 825	14,175
MNSZ2	28 777 590	444 760 553	15,455
Szószablya	24 991 306	462 024 888	18,487
huTenTen12	-	3 184 161 466	-

[4] Bár a módszer és az elkészült rendszer, képes más jellegű, illetve több független faktor együttes kezelésére, cikkünkben csak a fent említett faktorokat használtuk.

A mozaik n-grammokkal leírt minták elemzés nélkül, egészben történő feldolgozása analóg a számítógépeknél ismert *gyorsítótárazáshoz*, idegen szóval *cacheeléshez*. Az AnaGramma alapgondolata szerint, az emberi feldolgozás gyorsasága nagyrészt a gyakori esetek egészleges feldolgozásának tudható be, mivel az előre tárolt minták segítségével nagymértékben csökkenthető a mondatszintű elemzés komplexitása, aminek következtében javulhat az elemzőrendszer minősége.

A modul egy másik fontos feladata, hogy a gyakori szerkezetek ismeretében képes megjósolni az elemzés során a szerkezeti határokat. Mivel a gyakori szerkezetekből és a Grice-i maximák [2] alapján arra számít az elemző, hogy a szöveg folytatása kiszámítható és az előzetes tapasztalatoknak megfelel. Ennek köszönhetően az elemzőrendszer többi moduljának képes jelezni, hogy az elemzés valószínűleg elért egy szerkezeti határt, vagy éppen hogy egy nagyobb szerkezeti egység közepén tart. Így képesek vagyunk támogatni az elemzőrendszer azon moduljait, amelyek a különböző elemzési szintek szerkezeteit keresik (pl. szintaktikai elemző, NP/VP chunker). A rendszer előnye továbbá, hogy *nyelvi modellként* képes lesz számszerű becslést adni a soron következő szóra és/vagy kategóriára.

Munkánk során nagy hangsúlyt fektettünk arra, hogy az általunk létrehozott rendszer valós időben tudjon a keresett mintákra példákat adni nagyméretű korpuszokból trigramnál magasabb rendben is. Ez különösen fontos, ha figyelembe vesszük, hogy az elemzés komplexitása exponenciálisan nő a különböző kategóriák számával, valamint az elemzett mondat hosszával.

A fent említett minták keresése azért kevéssé kutatott téma, mert nagyon nagy tárhelyet igényel, illetve napjainkig a számítási kapacitások szűkösek voltak. A keresési tér nagyon rosszul skálázódik (lásd 2. táblázat), ezért szükséges számtalan premissza figyelembevétele, úgy hogy a zajokat csökkentsük a rendszerben, míg a keresett elemeket megtartsuk.

2. táblázat. A korpuszokban mért kifejezések mennyisége

	Szeged Korpusz 2		MNSZ1		MNSZ2	
	szó	WLT	szó	WLT	szó	WLT
1-gram	129 273	181 279	6 297 534	9 519 354	7 208 999	8 650 798
2-gram	578 642	2 962 756	57 770 805	236 296 463	73 919 408	299 373 602
3-gram	918 915	17 641 621	135 616 024	2 028 400 881	191 820 777	2 589 580 489
4-gram	1 019 316	67 750 636	184 815 630	10 241 065 746	280 556 568	12 498 795 104
5-gram	998 515	213 126 488	197 430 850	28 785 417 930	314 801 331	42 073 197 888
6-gram	946 278	618 181 519	192 819 805	88 556 351 842	310 102 954	131 117 731 010
7-gram	887 086	1 742 852 595	182 743 426	259 778 917 230	305 349 214	400 011 439 879
8-gram	826 405	4 825 618 452	171 459 179	731 213 387 722	289 872 274	1 179 148 233 622
9-gram	766 638	13 429 864 821	160 185 064	2 207 045 830 298	273 095 868	3 493 974 880 398

3. Kapcsolódó munkák

Korpuszbeli minták keresésével a *Mazsola* nevű rendszer [3] is foglalkozik, de az igei vonzatkeretek detekciójánál, az igék sajátosságaiból fakadóan, a szavak

sorrendje sokkal szabadabb, mint az általános, főleg főnévi csoportokat és gyakori szekvenciáikat tartalmazó szerkezeteké. További jellemvonása a módszernek, hogy szintaktikailag elemzett bemenettel dolgozik, mely feltétel a mi megközelítésünknek ellentmond.

A legközelebbi hasonló implementáció a *SRILM* [4] nevű eszközben megvalósított faktoros nyelvmodell [5], mely módszer használható bigramra, de legfeljebb trigramra. Magasabb nyelvmodell kezelésére viszont a magas erőforrásigény és futásidő miatt nem alkalmas. Ezért a tár és számítási kapacitások figyelembevételével eltérünk a faktoros nyelvi modellektől. Ahogy az emberi elemzés során is, mi is csak a gyakori szerkezetek vizsgálatára hagyatkozunk, nem kívánunk teljes modellt adni a nyelvhez. Ezzel a feladat a létező kapacitások határain belül tartható, de alapjaiban más megközelítésre van szükség.

A Sketch Engine [6] a napjainkban elérhető legátfogóbb korpuszkezelő rendszer, melynek a nyílt forráskódú változatát (NoSketchEngine [7]) használtuk a korpuszokban való keresésre, illetve az általunk megtalált mintákhoz példák és gyakorisági eloszlások generálásához. Számtalan funkciója között megtalálható az n-gramok generálása is, de sebességben tapasztalataink szerint közel azonos teljesítményt nyújt a mi rendszerünkkel, továbbá nem képes mozaikgramok generálására. Ezért szükségszerű volt egy saját rendszer fejlesztése, mely kiegészítőként szolgál ehhez a maga területén rendkívül hatékony az eszközhöz.

4. Módszerek

Alapvetően a Humor kódokkal [8] elemzett és a Szeged korpuszon [9] tanított PurePOS 2.1-gyel [10] egyértelműsített korpuszokon vizsgálódtunk és három faktort vettünk figyelembe, a szóalakot, a lemmát és a szófaji címkét. Egy token az ezekből alkotott hármasokkal (WLT) volt reprezentálva és a hagyományos csak szóalakot vagy csak címkét tartalmazó n-gramok helyett a szótöveket is és a három faktor tetszőleges kombinációit is megvizsgáltuk. A bevezetett kategoriális megkülönböztetések segítségével elénk tárulnak olyan valóban gyakori esetek is, melyek csak a kategóriájuknál fogva képezik gyakori minták részét. A modul képes további kategóriák kezelésére is. Így lehetnek akár szemantikai jellegű megszorítások is (élő, intézmény, nyelv stb.).

Egy külön mérésben kíváncsiak voltunk továbbá arra, hogy ha egy egyszerű binárisan eldönthető kérdéssel „*Főnévi csoport (NP)* vagy nem főnévi csoport része az adott szó?" géppel felcímkézett nagy méretű korpuszok esetében az NP-k és az egymás mellett álló NP-k (mivel ezek nincsenek megkülönböztetve az egyszerűség kedvéért) belső szerkezete milyen tipikus mintázatokat mutat.

4.1. A nagy adatok problémája

A nagy korpuszok gyakran esnek olyan hibába, hogy mivel nincs emberi kapacitás kézzel elvégezni az annotációt, ezért gépi eszközöket futtatnak rajta, amik a pipeline architektúra miatt, felnagyítják a már meglevő hibákat. Például a nagy

korpuszok vizsgálata során találtunk olyan esetet, amikor egy táblázat egyes mezői, amik számokat tartalmaztak, külön-külön mondatokká lettek alakítva egy szavas, számokból álló mondatokat alkotva. Ezért alaprendszerként két egyszerű n-gram alapú nyelvfelismerőt[5] futtatunk a korpuszok mondatain és ahol mindkettő magyar nyelvűnek ítélte az adott mondatot, azt tekintettük jó mondatnak. Ezzel a nagyon durva méréssel a korpuszok kb. 30%-át találtuk használhatónak a modellalkotásunk céljára. Tudjuk, hogy a nyelvfelismerők a rövid (3-4 szavas) mondatok esetében nem mindig rendelkeznek elég információval a döntéshez, ezért tévednek, ám az általunk készített modellek esetében a rövid mondatok nem rendelkeznek elég információval, így a kihagyásuk nem okoz problémát. A mérést finomítani szeretnénk a jövőben a fordításminőségbecslő algoritmusok [12] korpuszminőségbecslő algoritmussá alakításával.

4.2. Felhasznált eszközök és technikák

A fenti számítások (2. táblázat) alapján úgy találtuk, hogy memóriában tárolni nem tudjuk egyben a szükséges adatokat, ezért lemezre írva kell tárolni és ennek a kritériumnak megfelelően feldolgozni minden köztes adatot. Választásunk az egyszerűbb, standardabb feladatok esetén *Unix Coreutils* parancsaira mint a *sort*, *uniq*, *(e)grep*, stb. esett, mert ezek lemezorientáltan nagyon hatékonyak és több tíz éves fennállásuk óta sokszor sikeresen alkalmazták őket hasonló területeken. Míg a bonyolultabb számításokat az *AWK* nyelv különböző variánsaival végeztük, mivel előzetes méréseinkben úgy találtuk, hogy már kicsi korpuszméreten is kiemelkedően jól teljesítenek, a gyors prototípus alkotást lehetővé tevő szkript nyelvekhez (mint a Pyhton és Perl) képest megtartva a gyors változtatások lehetőségét. Az AWK nyelv különböző implementációira azért volt együttesen szükség, mert az Unicode karaktereken történő változtatásokat nem igénylő feladatok, az MAWK[6] variánssal sokkal gyorsabban futottak le, mint a szabványnak tekinthető GNU AWK-val[7], míg az utóbbi segítségünkre volt az Unicode kisbetűsítés gyors elvégzésében.

4.3. A Zipf-görbe vágása

A korpusznyelvészetben jól ismert Zipf-görbe [13], ami egy szó előfordulási gyakoriságának és a gyakorisági táblában levő rangjának függvénye. Ez a görbe „emberi szemmel nézve" nagyon hasonlít a reciprok függvény egy változatára. A főbb alaktani tulajdonságai megmaradnak akkor is, ha nem szavakon, hanem lemmákon, címkéken vagy éppen ezekből alkotott n-gramokon nézzük a gyakoriságokat. A görbe lecsengése minden esetben nagyon hosszú, ezért a nagyon magas számú *Hapax Legomenonok*, illetve nagyon ritka elemek tárolására, amik statisztikailag nem bírnak információtartalommal, nincs szükség. Ezért szükséges egy *alsó küszöb* meghatározása, amivel a keresési tér méretét csökkentjük. A célunk az

[5] langid.py (https://github.com/saffsd/langid.py) és A textCat nyelvfelismerőt [11]
[6] http://invisible-island.net/mawk/
[7] https://www.gnu.org/software/gawk/

ember fejében alkalmazásra készen álló gyakori szerkezetek megtartása, a passzív nyelvtudást nem kívánjuk modellezni, ezért szükséges egy *felső korlát* meghatározása, ami a gyakori, aktívan behelyettesíthető mintákat elválasztja a passzív nyelvismerettől. A fenti kettő küszöb megállapítása szükségszerűen automatikus kell, hogy legyen és a korpusz méretétől függetlenül azonos eredményt kell produkáljon. Továbbá a kísérletezésnek teret hagyva kellően finoman változtatható kell, hogy legyen.

Célszerűnek látszott a görbéhez egy megfelelő meredekségű érintő húzása, mely a szóban forgó számoktól függetlenül megadja azt a pontot, ahol a szavak gyakorisága és rangja a kívánt arányt éri el. Felső korlátnak heurisztikusan a 45 fokot választottuk, míg alsónak a 10 fokot választottuk kiindulásként. Tapasztalataink szerint az első az elemek kevesebb mint 1%-ánál metsz, míg az alsó korlát hozzávetőlegesen az elemek 50%-át tartja meg. Így a memória és tárbeli korlátainkba beleférünk.

A „görbe hasonlat" viszont csak akkor alkalmazható, amíg nem számítógépnek kell feldolgoznia az adatot, ugyanis közelebbről megnézve a görbét láthatjuk, hogy az nem folytonos és nem is görbe, hanem lépcsőzetes egyenes vonalakból áll. Ha élünk azzal a közelítéssel, hogy az egyes ugrásait a függvénynek összekötjük, ezzel nulla meredekségű egyenes vonalakból különféle meredekségű egyeneseket képezve, akkor is csak egy (szabálytalan) törtvonalat kapunk. Továbbá mivel nem áll végtelen adat rendelkezésünkre, ezért bizonyos helyeken az adathiány miatt, egynél nagyobb ugrások is gyakran előfordulnak, így tovább csúfítva a „görbénket". Az így kapott függvényt tovább kéne interpolálni, hogy az érintő numerikus deriválással kiszámolható legyen.

Ezért a görbe további alakítása helyett az érintő megkeresését egy egyszerű konvex lineáris programozási feladatra (minimalizálásra) redukáljuk. A kapott egyenesek meghatároznak egy félsíkot és a megfelelő meredekségű egyenes mint célfüggvény minimalizálásával (balra tolásával) megkapjuk azt a pontot, ahol „utoljára metszi a görbénket". Ezt a pontot egészre kerekítve kapjuk meg a keresett küszöb értéket, ahol vágni szeretnénk. Az eljárás előnye, hogy gyors és az adatok minőségétől és a görbe meredekségétől függetlenül működik.

4.4. A megtalált minták súlyozása

A kigyűjtött mintáknál sok olyan eset fordul elő, hogy azonos gyakorisággal egy adott minta többször több formában is előfordul. Ezeket összevonjuk egy mintává a legspecifikusabbat megtartva, hogy ne befolyásolják kedvezőtlenül az igazi minták rangját. A gondos eljárás ellenére, a taggelési hibák miatt kicsit eltérő gyakorisági számú, ám azonos minták is előfordulhatnak. Jelenleg, ezekkel a kis számuk és rendezetlenségük miatt nincs értelme foglalkozni. Többségük a korpusz hibáiból adódik.

A minták relevanciájának megállapításában az előfordulási gyakoriságuk mellett fontos szempont, hogy mekkora mértékben részei nagyobb mintáknak. Egy minta fontosságát nagy mértékben csökkenti, ha az gyakran része egy nagyobb szerkezetnek. Ezt oly módon vettük figyelembe, hogy a kisebb részminta előfordulási gyakoriságát csökkentettük az őt tartalmazó nagyobb minta súlyozott

gyakoriságával. Ezt a súlyozási technikát Frantzi et al. [14] c-value-nak nevezték el. A módszer lényege, hogy miután az összes, a feltételeinknek megfelelő n-gramot kigyűjtöttünk ($1 < \mathrm{n} < 12$), mindegyikre meghatároztuk a hozzá tartozó *c-value*-t, ami az adott n-gram *kifejezés voltára utaló mérőszám* mely pontosabb képet ad a gyakorisági értékeknél. Ez az érték a következő képlettel írható le:

$$C_{value(a)} = log_2|a|(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)) \quad (1)$$

, ahol

- a: a vizsgált kifejezésjelölt
- $f(a)$: a kifejezésjelölt gyakorisága
- $|a|$: a vizsgált kifejezés hossza
- $P(T_a)$: annak a gyakorisága, hogy a jelölt hányszor fordul elő hosszabb kifejezés részeként
- $f(b)$: az ilyen, hosszabb kifejezések száma

5. Eredmények

Kivettük a korpuszból a [PUNCT] POS-taget és a hozzátartozó lemmát, mert úgy találtuk, hogy a szempontunkból fölösleges és nem mond többet, mint a „szóalak". Illetve hasonlóan azokat a lemmákat is kiszűrtük, amelyek megegyeztek a szótővel, ezzel növelve a minták átláthatóságát és csökkentve az állapotteret. Ennek eredményeképpen a számszerű gyakoriságok megváltoztak, a fölösleges minták egy része eltűnt, de az egész rendszer viselkedése érdemben nem változott.

Az MNSZ 2.0 korpusz vizsgálatakor úgy találtuk, hogy az 5-nél hosszabb gyakori szerkezetek kizárólagosan a parlamenti doménből származtak és általános nyelvi információk nem voltak kinyerhetőek belőlük. Azt találtuk, hogy a doménektől nagyon erősen függenek az ilyen szerkezetek. A szaknyelvi zsargonok és fordulatok használata épp úgy, mint az etikett és az egyéb udvariassági konvenciók betartásával keletkező „részben merev" ismétlődő szerkezetek nagy eltéréseket okoznak a domének között a gyakori mintákban.

Ha egy gyakori mintákhoz megnézzük, a konkrét előfordulásokat, amit a 3. táblázat mutat, láthatjuk, hogy bár az első szó meghatározása [FN][NOM] – biztosabbat nem tudunk mondani róla – a szófaji címkék finomítása ebben az esetben kívánatos lenne, mert a példák alapján látható, hogy a főnevek egy speciális alosztálya statisztikailag szignifikánsan megfigyelhető. Kivételek persze akadnak elenyésző számban, de ezek még mindig elemezhetők a hagyományos elemzéssel, míg a meghatározott gyakori osztály előelemezhető és mintaként beilleszthető. Továbbá a példában szereplő *azt* egy speciális esete az „azt" szónak, ahol a „mondta , hogy" szerkezet következik és nincs szükség visszamenőleges koreferenciafeloldásra, ami úgy állapítható meg, hogy egy szóval előre tekintünk az aktuális állapothoz képest, hogy felismerjük a „gyorsítási lehetőséget". Vegyük észre továbbá, hogy bár elméleti lehetősége van, az „ezt" szó nem szerepel hasonló kontextusban a korpuszban.

3. táblázat. *„[FN][NOM] [FN|NM][ACC] lemma:mond , [KOT]”* mintákhoz tartozó szóalakok és azok előfordulási gyakorisága

Vizsgált minta					Gyakoriság
[FN][NOM]	**[FN|NM][ACC]**	**lemma:mond**	**,**	**[KOT]**	**11918**
úr	azt	mondta	,	hogy	906
úr	azt	mondja	,	hogy	304
törvény	azt	mondja	,	hogy	176
miniszterelnök	azt	mondta	,	hogy	168
miniszter	azt	mondta	,	hogy	158
asszony	azt	mondta	,	hogy	126
államtitkár	azt	mondta	,	hogy	118
ember	azt	mondja	,	hogy	117
kormány	azt	mondja	,	hogy	108
gábor	azt	mondta	,	hogy	104
istván	azt	mondta	,	hogy	102
viktor	azt	mondta	,	hogy	98
lászló	azt	mondta	,	hogy	97
péter	azt	mondta	,	hogy	97
ferenc	azt	mondta	,	hogy	91
túlzás	azt	mondani	,	hogy	86
...					

A második példán a 4. táblázatban látható gyakori mondat esetén csak a számot és személyt kell behelyettesíteni két esetben. Ha egy olyan „félig elemzett szerkezetet” tartunk készenlétben a memóriában, amiket csak ezekkel a kérdéses részekkel kell paraméterezni, a mondat többi elemzési lépését teljes egészében megspórolhatjuk a gyorsítótárazással, így növelve az elemzés sebességét.

A harmadik példánkban a 5. táblázatban látható, hogy a „gondolom , hogy” szerkezetek a mondatokban az esetek túlnyomó részében „azt”-al kezdődnek, habár a lehetséges névmások tárháza sokkal nagyobb, a korpusz a nyelvhasználat statisztikával ezt nem igazolja vissza. Ezzel elkerülhetjük, hogy az elemzőnkben fölöslegese eseteket is számba vegyünk, akkor ha azok „egyszerűbb elemzési heurisztikákkal” gyorsan megelemezhetők, kikerülve az elemzési „tévutakat” és a kombinatorikus robbanást.

Az utolsó példán a 6. táblázatban láthatjuk, hogy a triviális mintában a kötőszavak eloszlása mennyire eltérő, akkor ha NP-n belüli vagy pedig általánosan tekintett mintákról beszélünk. Ebből levonhatjuk azt a következtetést, hogy ha az NP elejét tudjuk detektálni, akkor „átállíthatjuk az agyunkat, egy másfajta elemzési módba”, ahol a triviális minták is másképpen viselkednek az NP-k végéig. Ezzel leszűkítve az állapotteret, gyorsítva az elemzést a tipikus szövegeken. Ez a viselkedés igazolni látszik az AnaGramma elemző elvét, miszerint a „különböző modulok elemzés közben hatással vannak egymásra, kommunikálnak”.

4. táblázat. *„lemma:köszön a lemma:figyelem .”* mintához tartozó szóalakok és azok előfordulási gyakorisága

Vizsgált minta				Gyakoriság
lemma:köszön	**a**	**lemma:figyelem**	**.**	**14582**
köszönöm	a	figyelmüket	.	7654
köszönöm	a	figyelmet	.	6762
köszönöm	a	figyelmét	.	142
köszönjük	a	figyelmüket	.	32
köszönjük	a	figyelmet	.	12
köszöni	a	figyelmüket	.	5
köszönöm	a	figyelmünket	.	3
köszönöm	a	figyelmeteket	.	2
köszöni	a	figyelmet	.	1
köszönjük	a	figyelmét	.	1

5. táblázat. *„[FN|NM][ACC] gondolom , hogy [HA]”* minta esetén az „[FN|NM][ACC]” címkéhez tartozó szavak és azok előfordulási gyakorisága

[FN|NM][ACC]	**7067**
azt	7056
ezt	8
.azt	1
amit	1
-azt	1

6. táblázat. *„[KOT] [DET] [MN][NOM] [FN][NOM]”* minta esetén a kötőszavak száma a teljes korpuszon, illetve az NP-k esetén

Általános szövegekben	gyakoriság	NP-ken belül	gyakoriság
hogy	185 102	és	6 236
és	27 556	illetve	489
mint	20 069	valamint	472
valami	17 791	azaz	33
de	16 480	és/vagy	30
illetve	13 072	avagy	27

6. Konklúzió

Munkánk során létrehoztunk egy olyan rendszert, amely szöveges korpuszból különböző faktorok kombinációinak segítségével képzett n-gramok gyakoriságából előállít mintákat. A mintákhoz lekérdezhetőek gyakorisággal együtt a teljesen kitöltött példák, amelyekre az adott minták illeszkednek. Ezen mintákból válogatott példákon bemutattuk, hogy a leendő elemző rendszer elemzéseit a minták gyorsítótárazásával gyorsítani tudjuk, az állapottér alkalmas leszűkítésével a tipikus szövegek esetében, ezzel utánozva az emberi agy gyorsaságát hasonló helyzetekben. Továbbá bemutattuk, hogy adott információk ismeretében, az elemző belső állapota átállítható egy specifikus almintacsoport elemzésére, ami nagyobb léptékekben (például ilyen a különböző doménbe tartozó esetleg roncsolt szövegek feldolgozása) könnyen megfigyelhető az emberi elemzőnél. Bár sok triviális minta is keletkezett, a létrejött minták számtalan ötletet adhatnak, a korpuszokon megfigyelhető jelenségek keresésére annak, aki hajlandó végigböngészni azokat.

Hivatkozások

1. Prószéky, G., Indig, B., Miháltz, M., Sass, B.: Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In: X. Magyar Számítógépes Nyelvészeti Konferencia, Szeged (2014) 79–88
2. Grice, H.P.: Logic and conversation. na (1970)
3. Sass, B.: "Mazsola" - eszköz a magyar igék bővítményszerkezetének vizsgálatára. In Váradi, T., ed.: Válogatás az I. Alkalmazott Nyelvészeti Doktorandusz Konferencia előadásaiból, Budapest, MTA Nyelvtudományi Intézet (2009) 117–129
4. Stolcke, A.: Srilm - an extensible language modeling toolkit. (2002) 901–904
5. Bilmes, J.A., Kirchhoff, K.: Factored language models and generalized parallel backoff. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2, Association for Computational Linguistics (2003) 4–6
6. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The sketch engine. In: Proceedings of the Eleventh EURALEX International Congress. (2004) 105–116
7. Rychlỳ, P.: Manatee/bonito-a modular corpus manager. In: 1st Workshop on Recent Advances in Slavonic Natural Language Processing, within MU: Faculty of Informatics Further information (2007) 65–70
8. Novák, A.: What is good Humor like? In: I. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, SZTE (2003) 138–144
9. 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
10. 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, Hissar, Bulgaria (2013) 539–545
11. Cavnar, W.B., Trenkle, J.M., et al.: N-gram-based text categorization. Ann Arbor MI **48113**(2) (1994) 161–175

12. Yang, Z.Gy., Laki, L.J., Prószéky, G.: Gépi fordítás minőségének becslése referencia nélküli módszerrel. In: XI. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Szegedi Egyetem (2015) 3–13
13. Zipf, G.K.: Human behavior and the principle of least effort. (1949)
14. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms:. the c-value/nc-value method. International Journal on Digital Libraries **3**(2) (2000) 115–130