Korpuszépítés és -feldolgozás learatott webes tartalomból

Kalcsó Gyula[1], Mihály Eszter[1], Szűcs Kata Ágnes[1]

[1]Országos Széchényi Könyvtár, Digitális Bölcsészeti Központ, Digitális Filológiai és Webarchiválási Osztály, 1014 Budapest, Szent György tér 4-5-6.
{kalcso.gyula, mihaly.eszter, szucs.kata}@oszk.hu

Kivonat: A cikk az Országos Széchényi Könyvtár webaratási tevékenységének eredményeképpen létrejött webarchívum korpuszépítési hasznosulási lehetőségeit mutatja be. A fókusz a tematikus gyűjtések által felépített archívumrészekből történő tematikus korpuszépítésen van. Ismerteti a szövegkinyerés eljárását, amelynek eredményeképpen a learatott WARC-fájlokból a WARCIO Python-könyvtár felhasználásával elsőként HTML, majd az ún. boilerplate-nek a jusText nevű, Pythonban írt eszköz segítségével történő eltávolításával nyers szöveg jön létre. Ismertet egy példaprojektet is, amelynek során az emtsv-vel elemzett szövegekből kinyert adatok vizualizációja történt.

1 Az OSZK webaratási tevékenysége

Az Országos Széchényi Könyvtár a 2000-es évektől foglalkozik born digital dokumentumokkal, előbb a Magyar Elektronikus Könyvtár, majd az időszaki kiadványok számait archiváló Elektronikus Periodika Archívum és Adatbázis és a képi dokumentumokat gyűjtő Digitális Képarchívum keretében. A webarchívum létrehozásának az igénye már korábban is felmerült, de csak a 2015-ös gyűjtőköri szabályzatában rögzítette feladatként a nemzeti könyvtár. A szükséges informatikai és személyi feltételek végül 2017-ben teremtődtek meg, és egy két és fél éves tesztidőszak után elkezdődött a magyar webtérben közzétett tartalmak egy részének időszakos lementése és gyűjteménybe szervezése. Az OSZK webarchívuma azzal a céllal jött létre, hogy reprezentatív képet nyújtson az egy adott időszakban nyilvánosan elérhető, a magyar közönségnek szánt és a kulturális örökség részét képező online tartalomkínálatról, a hungarikumok körébe tartozó elektronikus dokumentumokról. Az Országos Széchényi Könyvtárban folyó webarchiválást a Digitális Bölcsészeti Központ Digitális Filológiai és Webarchiválási Csoportja végzi az 1997. évi CXL. törvény 59/A. paragrafusa és a Kormány 626/2020. (XII. 22.) számú rendelete alapján, az OSZK Gyűjtőköri Szabályzatában definiált elvek szerint. Az OSZK Webarchívum gyűjtőkörébe tartozik a magyar webtérben létező vagy valaha létezett, nyilvánosan közzétett digitális tartalmak összessége, beleértve tehát azokat is, amelyek már az élő weben nem elérhetők, de valahol még megőrződtek.

A dokumentumok gyűjtése háromféle módon történik: válogatva a legfontosabb magyar webhelyekről, kiemelt eseményekhez kötődve a főbb hírforrásokból, illetve általános jelleggel a magyar webtérről. Szelektíven kerül gyűjtésre a tudományos,

kulturális, oktatási, közéleti jellegű tartalmak meghatározott köre. Az általános gyűjtés a .hu domén alatt regisztrált vagy egyéb doménhez tartozó, de magyar közönséget megcélzó nyilvános webhelyekre terjed ki. A webaratás (crawling) csupán azon szervereket érinti, ahonnan technikailag biztosítható a tartalom automatikus lementése. Az aratás során a könyvtár figyelembe veszi a begyűjtő szoftver számára az adott webhely tulajdonosa által beállított korlátozásokat.

Az archivált webtartalom esetében a nemzeti könyvtár elsősorban annak hosszú távú megőrzésére törekszik. A szerzői és személyiségi jogok tiszteletben tartása érdekében a gyűjteménynek csak egy kis része tekinthető meg nyilvánosan, olyan webhelyek, amelyeknél a tartalom tulajdonosa erre engedélyt adott. Az archívum többi része csak a könyvtár zárt szolgáltatási felületén elérhető – elsősorban kutatási célokra.

A nemzeti könyvtár a webtérszintű és a tematikus aratások mellett eseményalapú gyűjteséket is végez a jelentősebb kulturális, politikai és sporteseményekről. 2022. február 21. óta elkezdte gyűjteni az orosz–ukrán konfliktussal majd később háborúval kapcsolatos híreket, 75 magyarországi és határon túli hírportálról. A hírek gyűjtése alapvetően a portálokon használt címkék vagy kategóriák alapján történik (ez 445 seed-URL-t jelent). Ezek a mentések hetente egyszer futnak. A gyűjtemény nem nyilvános, azonban a SolrWayback nevű szoftver segítségével készült hozzá egy nyilvános kereső, mellyel a hírek nem nézhetők meg szerzői jogi okok miatt, de a teljes szövere lehet keresni, a metaadatok illetve a szövegkörnyezet megjeleníthető. Március elejétől egy másik tematikus gyűjtemény is létrejött olyan webhelyekről és közösségimédia-oldalakról, melyek a Kárpátalján élő magyarok számára fontosak, beleértve a segítségnyújtással foglalkozó Facebook-csoportokat is. Ez több mint 1000 seed URL-t tartalmaz, s hetente egy aratás gyakorisággal archiválódik. A Digitális Bölcsészeti Központ a hírportálok anyagából épített tematikus korpuszt, a cikk ennek a munkálatait ismerteti a továbbiakban.

2 Előzmények: magyar webkorpuszok

Bár a tematikus korpuszépítés nemzetközi szinten is bevett gyakorlat (l. Barbaresi 2019), az ún. boilerplate removing (l. 3.2) nyelvspecifikussága miatt itt csak a magyar előzményeket tekintjük át. A korábban épült, legalább részben webes forrásokon alapuló korpuszokat Indig (2018) ismerteti (Indig, 2018: 127). Ebben öt korpuszról történik említés, amelyek valójában három projekt keretében készültek el (Webkorpusz, Magyar nemzeti szövegtár, Pázmány-korpusz), az első és a második két verzióban. Ezek módszertana bizonyos mértékig tekinthető mintának, azonban a webaratás (crawling) technikai feltételeinek a változásai, valamint az újabb webes technológiák megjelenése miatt csak részben használható fel újabb korpuszok építésekor.

Maga Indig (2018) is egy módszertant ismertet, amely arra irányul, „hogy a szabadon hozzáférhető anyagból milyen mennyiségű és minőségű magyar nyelvű korpusz gyártható” (Indig 2018: 129). A módszer az alábbi lépésekben nyer ki szöveget a CommonCrawl archívumából: a WARC-indexből kinyeri a letöltendő oldalak adatait (előzetesen kiszűrve a korpuszépítés szempontjából irreleváns tartalmakat), letölti az

oldalt, és azonnal megpróbálja belőle kinyerni a szöveget a jusText Python-eszköz segítségével.

A legfrissebb webkorpusz Nemeskey Dávid Webkorpusz 2.0 projektje keretében készült el (Nemeskey 2020). Ennek mérete messze meghaladja a korábbiakét (több mint 9 milliárd token). Nemeskey a korpusz letöltéséhez és feldolgozásához saját Python-eszközt is készített, amely szabadon elérhető a Githubon[1]. A szöveg kinyeréséhez ez a projekt is a jusTextet használta. További érdekessége, hogy a letöltött szövegeket utólag szűrte (pl. az 1500 szónál rövidebb vagy nem magyar nyelvű tartalmakat eldobta). Ugyancsak fontos sajátossága, hogy többféle duplikátumszűrést is végzett. Dokumentumszinten az ún. minhash algoritmus segítségével távolította el az egyező tartalmakat. Ezen túl azonban a gyakori egyező szövegrészeket (bekezdéseket) is kiszűrte korpusz- és dokumentumszinten is.

3 A szöveg kibontása a learatott webes tartalomból

Az OSZK által szelektíven learatott webes tartalomból más jellegű korpuszok építhetők, mint a korábbi projektek főként CommonCrawl alapú korpuszai. A szelektív aratás módszertana lehetővé teszi, hogy az archivált tartalomból tematikus korpuszok épüljenek.

3.1 A HTML-tartalom kinyerése a WARC-okból

A korpuszépítés első lépéseként a learatott WARC-fájlokból kellett kinyerni a HTML-részeket, amelyek a szöveget is tartalmazzák. Ehhez a WARCIO nevű Python-könyvtárat használtuk, amely a Webrecorder szoftver része[2]. Ennek segítségével a WARC-okban tárolt bármely MIME-típusú tartalom kibontható. Az alábbi táblázat mutatja, hogy a február 21. és június 6. között learatott tartalomból mennyi HTML-t bontott ki a script:

1. táblázat: A korpusz alapját képező learatott webes tartalom főbb adatai

Dátum	WARC-ok száma (mérete)	HTML-ek száma (mérete)
2022-02-21	12 (8,79 GB)	82906 (8,94 GB)
2022-02-28	11 (6,73 GB)	77913 (8,66 GB)
2022-03-07	12 (6,19 GB)	78737 (8,69 GB)
2022-03-14	12 (5,74 GB)	80500 (8,86 GB)
2022-03-21	12 (5,2 GB)	80806 (9,25 GB)
2022-03-28	13 (7,12 GB)	102096 (11,14 GB)
2022-04-05	8 (7,43 GB)	97427 (11,12 GB)
2022-04-11	15 (6,45 GB)	95666 (11,18 GB)

[1] https://github.com/DavidNemeskey/cc_corpus
[2] https://github.com/webrecorder/warcio

2022-04-18	13 (6,11 GB)	94260 (11,09 GB)
2022-04-25	13 (6,32 GB)	100361 (12,09 GB)
2022-05-02	14 (7,14 GB)	98847 (12,11 GB)
2022-05-09	13 (6,7 GB)	100667 (12,1 GB)
2022-05-16	14 (6,63 GB)	105468 (12,67 GB)
2022-05-23	13 (6,34 GB)	102542 (12,27 GB)
2022-05-30	13 (6,34 GB)	101333 (12,1 GB)
2022-06-06	17 (7,19 GB)	105505 (12,9 GB)

3.2 A jusText algoritmus

A következő lépésben a HTML-fájlokból kellett kinyeri a szöveget. Az ún. boilerplate removal a jusText algoritmus (Pomikálek 2011) segítségével történt[3]. Az algoritmus a HTML egyszerű szegmentálása alapján működik. Egyes HTML-tagek tartalmát a webböngészők (alapértelmezés szerint) vizuálisan blokkokként formázzák. Az ötlet lényege, hogy a script szöveges blokkokra szegmentálja a tartalmat a tagek alapján. A használt blokkszintű tagek teljes listája a következő: `blockquote`, `caption`, `center`, `col`, `colgroup`, `dd`, `div`, `dl`, `dt`, `fieldset`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `legend`, `li`, `optgroup`, `option`, `p`, `pre`, `table`, `td`, `textarea`, `tfoot`, `th`, `thead`, `tr`, `ul`. A blokkokat két vagy több `br` tagből álló rész is elválaszthatja egymástól.

Bár az ilyen blokkok némelyike tartalmazhatja a releváns tartalom és boilerplate keverékét, ez meglehetősen ritka. A legtöbb blokk ebből a szempontból homogén.

Az ilyen blokkokkal kapcsolatban több megfigyelés is tehető:

- A linket tartalmazó rövid blokkok szinte mindig boilerplate jellegűek.
- Minden olyan blokk, amely sok linket tartalmaz, majdnem mindig boilerplate.
- A nyelvileg magas szinten strukturált (pl. mondatokból álló) szöveget tartalmazó hosszú blokkok szinte mindig jók, míg az összes többi hosszú blokk szinte mindig boilerplate.
- Mind a jó (fő tartalom), mind a boilerplate blokkok hajlamosak tömbökben előfordulni, azaz egy boilerplate blokkot általában más boilerplate blokkok vesznek körül, és fordítva.

A jusText azt, hogy egy szöveg nyelvileg magas szinten strukturált vagy nem, egyszerű heurisztikával, a funkciószavak (stopszavak) mennyisége alapján állapítja meg. Míg egy nyelvileg magas szinten strukturált szöveg jellemzően tartalmaz bizonyos arányban funkciószavakat, addig az olyan egyszerű tartalmakban, mint a listák és felsorolások, kevés funkciószó lesz. A script beépítetten tartalmazza a használható nyelvek stopszólistáit (többek között a magyarét is).

[3] Bár létezik a publikált adatok alapján hatékonyabb, ráadásul magyar fejlesztésű algoritmus a problémára (Endrédy–Novák 2013), a forráskód hozzáférhetetlensége miatt le kellett mondanunk a használatáról.

Az algoritmus kulcsgondolata, hogy a hosszú blokkokat, valamint bizonyos rövid blokkokat nagyon nagy megbízhatósággal lehet osztályozni. Az összes többi rövid blokkot ezután a környező blokkok vizsgálatával lehet osztályozni.

3.2.1 Előfeldolgozás

Az előfeldolgozás fázisában a `<header>`, `<style>` és `<script>` tagek tartalma törlésre kerül. A `<select>` tagek tartalmát boilerplate-ként jelöli meg a script. Ugyanez igaz a copyright szimbólumát (©) tartalmazó blokkokra.

3.2.2 Kontextusfüggetlen osztályozás

A szegmentálás és az előfeldolgozás után kontextusfüggetlen osztályozás történik, amelynek során minden blokk besorolódik az alábbi négy osztály valamelyikébe:

rossz (bad) – boilerplate blokk
jó (good) – releváns tartalommal bíró blokk
rövid (short) – túl rövid ahhoz, hogy megbízhatóan oszályozható legyen
majdnem jó (near-good) – a két utóbbi között

Az osztályozás az alábbi algoritmus alapján történik:

```
if link_density > MAX_LINK_DENSITY:
    return 'bad'

# short blocks
if length < LENGTH_LOW:
    if link_density > 0:
        return 'bad'
    else:
        return 'short'

# medium and long blocks
if stopwords_density > STOPWORDS_HIGH:
    if length > LENGTH_HIGH:
        return 'good'
    else:
        return 'near-good'
if stopwords_density > STOPWORDS_LOW:
    return 'near-good'
else:
    return 'bad'
```

A hosszúság karakterszámban van mérve. A linksűrűség az <a> tagekben szereplő karakterek arányával van meghatározva. A stopszavak sűrűsége a stopszólistán szereplő szavak arányával van meghatározva.

Az algoritmus két egész számot (LENGHT_LOW és LENGTH_HIGH), valamint három tizedestörtet (MAX_LINK_DENSITY, STOPWORDS_LOW és STOPWORDS_HIGH) használ paraméterekként. Az első kettő használatos a blokkok rövid, közepes és hosszú kategóriákba sorolásához. Az utolsó három pedig a stopszósűrűség alacsony, közepes és magas mértékének a megállapításához használatos. Az alapértelmezett beállítások a következők:

MAX_LINK_DENSITY = 0,2
LENGTH_LOW = 70
LENGTH_HIGH = 200
STOPWORDS_LOW = 0,30
STOPWORDS_HIGH = 0,32

A közepes és hosszú blokkok osztályozását a blokkméret és a stopszósűrűség alapján a következő táblázat foglalja össze:

2. táblázat: A közepes és hosszú blokkok osztályozása a blokkméret és a stopszósűrűség alapján

Blokkméret (szószám)	Stopszósűrűség	Osztály
közepes	alacsony	rossz
hosszú	alacsony	rossz
közepes	közepes	majdnem jó
hosszú	közepes	majdnem jó
közepes	magas	majdnem jó
hosszú	magas	jó

3.2.3 Kontextusfüggő osztályozás

Az algoritmus kontextusérzékeny részének célja, hogy a rövid és majdnem jó blokkokat a környező blokkok osztályai alapján jónak vagy rossznak minősítse át. A már jónak vagy rossznak minősített blokkok ebben a szakaszban viszonyítási pontként szolgálnak.

Az előzetesen besorolt blokkok rövid és majdnem jó blokkok jó és rossz blokkokkal határolt tömbjeinek tekinthetők. Minden ilyen sorozatot két jó blokk, két rossz blokk vagy egyik oldalon egy jó blokk, a másikon pedig egy rossz blokk övezhet. Az előbbi két eset könnyen kezelhető. A szekvencia minden blokkja jónak, illetve rossznak minősül. Az utóbbi esetben a rossz blokkhoz legközelebbi, majdnem jó blokk szolgál a jó és rossz terület elhatárolására. A rossz és a közel jó blokk közötti összes blokk rossznak minősül. Az összes többi jónak minősül. Ha a tömbben az összes blokk rövid (nincs közel jó blokk), akkor a blokkok mindegyike a jó blokkok közé sorolódik.

A kibontott szövegekben összességében elenyészően kis arányban maradt boilerplate jellegű tartalom. A paraméterezéssel és a magyar stopszólista felülvizsgálatával a script teljesítménye a későbbiekben talán még javítható. Az egyes aratásokból kibontott nyers szöveges tartalom főbb adatai a következők:

3. táblázat: Az egyes aratásokból kibontott szövegek főbb adatai

Dátum	TXT mérete	Tokenszám (kerekítve)
2022-02-21	199 MB	26 millió
2022-02-28	204 MB	26 millió
2022-03-07	206 MB	27 millió
2022-03-14	206 MB	27 millió
2022-03-21	209 MB	26 millió
2022-03-28	260 MB	31 millió
2022-04-05	241 MB	29 millió
2022-04-11	253 MB	31 millió
2022-04-18	257 MB	31 millió
2022-04-25	269 MB	33 millió
2022-05-02	261 MB	34 millió
2022-05-09	269 MB	35 millió
2022-05-16	285 MB	35 millió
2022-05-23	280 MB	34 millió
2022-05-30	269 MB	33 millió
2022-06-06	271 MB	33 millió

4 Adatvizualizáció a korpuszból

Jelen cikknek nem célja, hogy a korpuszból kibányászott adatokat elemezze, csupán azt mutatja be, hogyan tettük lehetővé a szövegek feldolgozását. A korpusz első adatvizualizációs projektje egy interaktív felület elkészítése volt, amely a magyar internetes sajtó orosz–ukrán háborúval kapcsolatos szóhasználatának a változását volt hivatott illusztrálni. Ennek elkészítéséhez először nyelvi elemzés készült az emtsv (Váradi et al. 2017) segítségével. A pipeline-ból a tok, morph, spell, pos modulokra volt szükség, azaz tokenizáltunk, morfológiailag elemeztünk (lemmatizálással), és POS-tagginget végeztünk. Az elemzésből elkészült a szótövek gyakorisági listája, amelynek a felső 2000 elemét automatikus és kézi eszközökkel megtisztítottunk (pl. automatikusan eltávolítottunk bizonyos, a szófelhő szempontjából lényegtelen szófajokat, mint pl. a névutók, valamint kézzel kivettünk a szövegekben maradt boilerplate jellegű tartalomból származó szavakat, „szemetes” tartalmat)[4].

[4] A szűrt szólisták letölthetők az oldal alján található linkekről: https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet

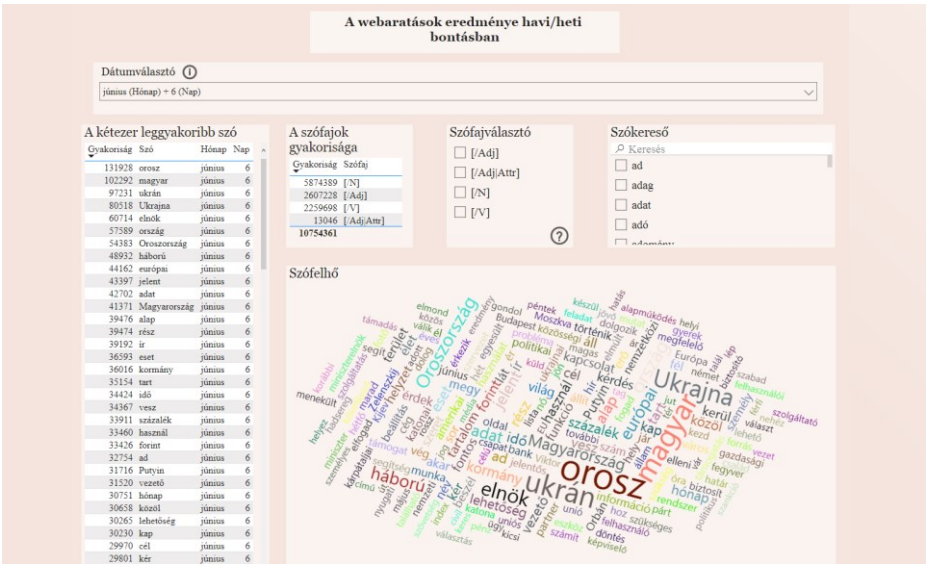

Fig. 1. Az orosz–ukrán háború cikkanyagából épített korpusz adatvizualizációja a Power BI felületén (https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet-interaktiv).

Az interaktív vizualizációt a magyar hírportálok orosz–ukrán háborúról szóló cikkeinek szóhasználatáról a dhupla.hu kreatív oldalán tettük közzé böngészhető, lekérdezhető, adatokat testre szabható formában[5]. Az adatok megjelenítéséhez a Microsoft által üzemletetett, ingyenesen is hozzáférhető Power BI szolgáltatást használtuk. Az alapvető koncepció az volt, hogy a szólisták manuális tisztítása után a kétezer leggyakoribb szót a webaratások heti bontásában egyesével és összesítve is meg tudjuk jeleníteni. A nyitóoldalon találhatóak az Összesített adatok. A Szófelhő az összesített adatok alapján méretezi a szavakat: minél nagyobb, annál több előfordulással szerepel. Az Idővonal segítségével heti vagy havi bontásban lehet szűrni, illetve az egymás melletti mezőket összevonva is lehet egyéni szűréseket végezni. Mellette táblázatos formában tudjuk nyomon követni a szavak alakulását: gyakoriság, hónap-nap és szófaj szerint. Az összesített adatok másik szűrési lehetőségét egy szókereső mező adja, ahol a listában szereplő szavakra egyesével is rá lehet keresni. A szavakat gyakoriság szerint teszi sorba egy diagram az idővonal alatt, melyet Egy statisztika és egy információs panel követ. A második oldalon egyesével tudjuk elemezni, vizsgálni a heti és a havi webaratások szókészletét. A dátumválasztó segítségével egy teljes hónapra vagy egy-egy hétre lehet szűrni. A szavak táblázatos formában gyakoriság és szófaj szerint rendezhetők, illetve szó és szófaj szerint is külön szűrhetők. Az oldal közepén lévő szófelhő interaktivitását az adja, hogy a táblázatok és a vizualizációk egymásra hivatkoznak. Így, ha valamelyik táblázaton szűrést végzünk, annak eredménye automatikusan látszik az ábrákon is. Háromféle megjelenítést ábrázolja a szavakat; a szófelhőn kívül egy sávdiagram mutatja a szavak gyakorisági sorrendjét és a fatérkép pedig a szófajok alakulását teszi láthatóvá az egyes webaratások adatain. Az oldalakat

[5] https://dhupla.hu/page/kreativ/ukrajna-hirek-szokeszlet-interaktiv

elláttuk olyan plusz információkkal is, amelyek a használatban és az eligazodásban segítik a felhasználókat.

5 Összegzés, kitekintés

A cikk az Országos Széchényi Könyvtár webarchívumának módszertanát mutatta be, amelynek segítségével a learatott webes tartalomból (WARC-fájlokból) nyelvileg elemzett tematikus korpuszt lehet építeni. Az eljárás további finomításával lehetséges tetszőleges, a webarchívumban mentett szövegegyüttesből korpuszok építése, valamint a nyelvileg elemzett szövegek adat-/szövegbányászati feldolgozása. A rugalmas korpuszépítéshez bizonyos problémákat még meg kell oldani: terveink közt szerepel a webarchívum hatalmas szövegmennyiségének automatikus tárgyszavazása (topic modeling), valamint fontos metaadatok (pl. a megjelenési dátum) automatikus szöveghez rendelése. Mindezek alapján lehetségessé válhat különböző témájú részkorpuszok építése akár egy bizonyos időszakra vonatkozóan is, ami a webarchívum bölcsészet- és társadalomtudományi kutatási hasznosulását nagyban elősegítené. Problémát jelent ugyanakkor, hogy a jogvédett szöveges tartalom (pl. a hírportálokról aratott szöveg) nem publikálható, megfelelő felhasználói felület létrehozásával biztosítani kell, hogy az adat-/szövegbányászat elvégezhető legyen.

Bibliográfia

Barbaresi, A.: The Vast and the Focused: On the need for thematic web and blog corpora. 7th Workshop on Challenges in the Management of Large Corpora (CMLC-7), Jul 2019, Cardiff, United Kingdom. pp. 29–32. (2019)

Indig B., Sass B., Simon E., Mittelholcz I., Kundráth P., Vadász N.: emtsv – Egy formátum mind felett. In: Berend Gábor, Gosztolya Gábor, Vincze Veronika (szerk.): MSZNY 2019, XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, 235–247. (2019)

Endrédy, I., Novák, A.: More Effective Boilerplate Removal—The GoldMiner Algorithm. Polibits – Research Journal on Computer Science and Computer Engineering with Applications, 48, 79–83. (2013)

Endrédy, I., Prószéky, G.: A Pázmány Korpusz. Nyelvtudományi Közlemények, 112. pp. 191–205. (2016)

Indig, B.: Közös crawlnak is egy korpusz a vége – Korpuszépítés a CommonCrawl .hu domainjából. In: XIV. Magyar Számítógépes Nyelvészeti Konferencia : MSZNY 2018. pp. 125–134. Szegedi Tudományegyetem, Szeged (2018)

Indig, B., Kákonyi, T., Novák, A.: Crawling in Reverse – Lightweight Targeted Crawling of News Portals. In: Kubis, M. (szerk.): Proceedings of the 9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. pp. 81–87. (2019).

Indig, B., Knap, Á., Sárközi-Lindner, Zs., Timári, M., Palkó, G.: The ELTE.DH Pilot Corpus. In: Barbaresi, A. (szerk.): Proceedings of the LREC 2020 12th Web as Corpus Workshop (ACL SIGWAC) pp. 33–41. (2020)

Nemeskey, D. M.: Natural Language Processing methods for Language Modeling. PhD thesis. Eötvös Loránd University (2020)

Pomikálek, J.: Removing Boilerplate and Duplicate Content from Web Corpora. PhD thesis. Masaryk University, Brno (2011)

Váradi, T., Simon, E., Sass, B., Gerőcs, M., Mittelholcz, I., Novák, A., Indig, B., Prószéky, G., Farkas, R., Vincze, V.: Az e-magyar digitális nyelvfeldolgozó rendszer. In: MSZNY 2017, XIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport, pp. 49–60. (2017)

Tagmondatokra bontás és NP-chunking függőségi alapon

Dömötör Andrea[1,2,3], Nemeskey Dávid[1,2]

[1]ELTE BTK TI Digitális Bölcsészet Tanszék
1088 Budapest Múzeum krt. 6-8.
[2]Digitális Örökség Nemzeti Laboratórium
[3]PPKE BTK Nyelvtudományi Doktori Iskola
1088 Budapest, Mikszáth Kálmán tér 1.
{domotor.andrea,nemeskey.david}@btk.elte.hu

Kivonat Ebben a cikkben a tagmondatokat és a köztük lévő kapcsolat típusát a függőségi elemzés mintázataiból kíséreljük meg meghatározni. Mivel ennek a feladatnak a teszteléséhez még nincs gold sztenderd adatunk, a módszerünket kipróbáltuk egy másik feladaton, az NP-chunkingon is. Ez utóbbi kiértékelésénél nehézséget okozott, hogy az elvben gold sztenderd korpuszok több hibát is tartalmaztak, mind a függőségi elemzésben, mind az NP-chunkingban. Mindezekkel együtt 89%-os f-score-t értünk el, ami ugyan elmarad a state-ot-the-arttól, de abból a szempontból mégis ígéretes, hogy ezt az eredményt egy egyszerű szabályrendszerrel értük el. Ez alapján a függőségi elemzés mintaillesztése további kutatásra érdemes módszer lehet a hasonló feladatokban. **Kulcsszavak:** NP-chunking, függőségi elemzés, tagmondatok, mintaillesztés

1. Bevezetés

A legtöbb korpusz és nyelvfeldolgozó eszköz a szöveget mondatok, azon belül pedig tokenek összességeként kezeli. A mondat és a token szintje között lehetnek még többszavas kifejezések (chunkok), azonban a tagmondat mint nyelvi szint jellemzően nem jelenik meg a számítógépes szövegfeldolgozásban.

Az összetett mondatok tagmondatai teljes értékű mondatoknak tekinthetők, így elkülönítésük hasznos lehet olyan feladatok esetén, ahol fontosak a (tag)mondathatárok, ilyen lehet például az elváló igekötők és igéik összekapcsolása. Ezen kívül a tagmondatok közötti viszonyok meghatározóak a szövegértelmezés szempontjából: egy kötőszón múlhat, hogy egy szövegrész tartalma egy másikéból következik-e vagy fordítva.

Ebben a cikkben a tagmondatokat és a köztük lévő kapcsolat típusát a függőségi elemzés mintázataiból kíséreljük meg meghatározni. Mivel ennek a feladatnak a teszteléséhez még nincs gold sztenderd adatunk, a módszerünket kipróbáltuk egy másik feladaton, a főnévi csoportok felderítésén (*NP-chunkingon*) is. Bár itt természetesen két eltérő feladatról van szó, látni fogjuk, hogy mindkettő megoldható függőségi mintaillesztéssel.