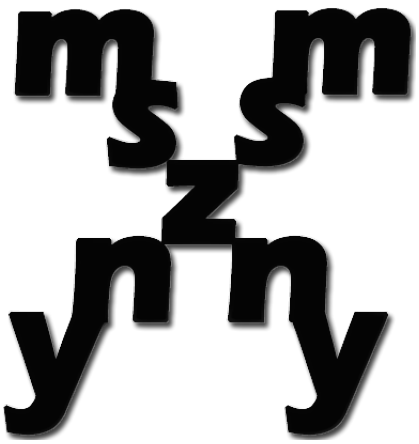
XIX. Magyar Számítógépes Nyelvészeti Konferencia



Szerkesztette:
Berend Gábor
Gosztolya Gábor
Vincze Veronika

Szeged, 2023. január 26–27.

Szerkesztette:
Berend Gábor, Gosztolya Gábor, Vincze Veronika
{berendg,ggabor,vinczev}@inf.u-szeged.hu

Felelős kiadó:
Szegedi Tudományegyetem
TTIK, Informatikai Intézet
6720 Szeged, Árpád tér 2.

ISBN: 978-963-306-912-7

Nyomtatta:
Innovariant Nyomdaipari Kft.
6750 Algyő, Ipartelep 4.

Szeged, 2023. január

Az MSZNY 2023 konferencia szervezője:
ELKH-SZTE Mesterséges Intelligencia Kutatócsoport

Előszó

2023. január 26–27-én már tizenkilencedik alkalommal kerül sor a Magyar Számítógépes Nyelvészeti Konferencia megrendezésére. Ebben az évben először hibrid formátumban kerül lebonyolításra a konferencia, lehetőséget teremtve mind a személyes részvételre, mind pedig a konferencia élő közvetítésének nyomon követésére.

A konferencia fő célkitűzése a kezdetek óta állandó: lehetőséget biztosítani a nyelv- és beszédtechnológia területén végzett kutatások eredményeinek ismertetésére és megvitatására, ezen felül a különféle hallgatói projektek, illetve ipari alkalmazások bemutatására. A hagyományokat követve a konferencia idén is nagyfokú érdeklődést váltott ki az ország nyelv- és beszédtechnológiai szakembereinek körében. A 43 beküldött cikkből gondos mérlegelést követően 35 cikk került elfogadásra, melyek témája számos szakterületre terjed ki a legújabb nyelvi modellek bemutatásától kezdve a beszédtechnológia eredményein keresztül a gépi fordításig.

Nagy örömet jelent számunkra, hogy Tikk Domonkos elfogadta meghívásunkat, aki plenáris előadását *A Netflix Prizetól a Tabooláig: a Gravity és az ajánlórendszerek fejlődése* címmel fogja megtartani.

Az idei évben is különdíjjal jutalmazzuk a konferencia legjobb cikkét, mely a legjelentősebb eredményekkel járul hozzá a magyarországi nyelv- és beszédtechnológiai kutatásokhoz. Ezen felül immár ötödik alkalommal osztjuk ki a legjobb bíráló díját, amellyel a bírálók fáradságos, ugyanakkor nélkülözhetetlen munkáját kívánjuk elismerni.

Köszönettel tartozunk továbbá a Neumann János Számítógéptudományi Társaság Mesterséges Intelligencia Szakosztályának a konferencia sikeres lebonyolításához nyújtott anyagi hozzájárulásáért.

A szervezőbizottság nevében,
Ács Judit, Berend Gábor, Gosztolya Gábor, Nemeskey Dávid Márk, Novák Attila, Simon Eszter, Sztahó Dávid, Vincze Veronika

Tartalomjegyzék

Mondatszám-meghatározás hatása a magyar nyelvű jogi szövegek extraktív kivonatainak minőségére

Csányi Gergely Márk[1], Gadó Krisztián[1], Bajári Lúcia[1], Megyeri Andrea[2], Fülöp Anna[2], Egri Erika[2], Vági Renátó[1,3], Nagy Dániel[1], Vadász János Pál[1,4], Üveges István[1,5]

[1]MONTANA Tudásmenedzsment Kft., 1029 Budapest, Hársalja utca 32.
[2]Wolters Kluwer Hungary Kft., 1117 Budapest, Budafoki út 187-189.
[3]Eötvös Loránd Tudományegyetem Állam- és Jogtudományi Doktori Iskola, 1053 Budapest, Egyetem tér 1–3.
[4]Nemzeti Közszolgálati Egyetem, Információs Társadalom Kutatóintézet, 1083 Budapest, Ludovika tér 2.
[5]Szegedi Tudományegyetem Nyelvtudományi Doktori Iskola, 6722 Szeged, Egyetem utca 2.
{csanyi.gergely,gado.krisztian,bajari.lucia,vagi.renato,
nagy.daniel,vadasz.pal,uveges.istvan}@montana.hu
{andrea.megyeri,anna.fulop}@wolterskluwer.com

Kivonat Az egyes dokumentumok tartalmi összefoglalása során a cél egy dokumentum rövidebb változatának előállítása úgy, hogy annak fő információtartalma a kivonatban megőrződjön. Cikkünkben az anonimizált bírósági határozatokhoz készült extraktív kivonatoló rendszer fejlesztése során szerzett tapasztalatokat ismertetjük, különös tekintettel a kivonatok hosszával (mondatszám) kapcsolatban felmerült kérdésekre, és az azokra adott válaszainkra. A kivonatokkal egy jogi adatbázis felhasználóinak találati listában való könnyebb orientációját kívántuk támogatni.
Kulcsszavak: extraktív szövegösszefoglalás, kivonatolás

1. Bevezetés

Egy szöveg kivonatán a dokumentum tartalmának lehetőség szerint rövid, de pontos reprezentánsát értjük. Módszertanát tekintve beszélhetünk extraktív vagy absztraktív szövegösszefoglalásról attól függően, hogy az adott szöveg tartalmi kivonatát milyen módszerrel állítjuk elő. Extraktív esetben a kivonat részét képező mondatok szó szerint, az eredeti szövegből kiemelve kerülnek a kivonatba, míg absztraktív esetben a cél olyan szintaktikaliag és szemantikailag is helyes mondatok generálása, amelyek ugyancsak helyes összefoglalását adják a kiindulási szövegnek (Nenkova és McKeown, 2011).

A megfelelően elkészített kivonatokkal jelentősen javítható például jogi keresőrendszerek teljesítménye azáltal, hogy egy keresésre válaszul nem pusztán relevánsnak ítélt dokumentumok egy halmazát, hanem azok tartalmi összefoglalását

is visszaadjuk. Az ilyen összefoglalások megfelelő választ nyújthatnak például a jogi keresőket gyakran jellemző homogenitás problémájára, amikor is a visszakapott találati listákban a technikai adatokon túl nem jelenik meg olyan érdemi információ, amely alapján a felhasználó a dokumentumba való beleolvasás nélkül is megítélhetné annak tényleges relevanciáját (Vági, 2022).

A célunk egy olyan modul elkészítése volt, amely által előállított kivonatok a fent említett módon képesek orientálni a felhasználókat egy jogi keresőrendszer használata közben. A konkrét implementációt a bírósági határozatok kivonatolásának automatizálása céljából valósítottuk meg. A kivonatkészítést extraktív alapokon készítettük el, mivel elvárás volt, hogy a kivonatok csakis nyelvileg helyes mondatokat tartalmazzanak. Felügyelet nélküli gépi tanulással kísérleteztünk, tekintettel arra, hogy a megbízhatóbb megoldások ezek között találhatók (Schluter, 2017).

Általános esetben egy kivonat minőségét felügyelet nélküli, extraktív kivonatolás esetén befolyásolhatjuk az abba kerülő lehetséges szövegegységek rangsorolási módjának változtatásával, a vektorizálási forma megválasztásával, a kivonat szempontjából nem releváns szövegrészek kiszűrésével, valamint a kivonat hosszának változtatásával is. Míg előbbi három viszonylag gyakran előkerülő kérdéskör a szakirodalomban, addig az utóbbi azonban egy jelentősen alulreprezentált problémakör, ezért is szenteltünk különös figyelmet a vizsgálatának. A kivonatok hossza pedig amiatt is különösen releváns, mivel az üzleti környezetben működő rendszerek esetében gyakran elvárás lehet az ezekre vonatkozó adott karakterlimit betartása (Mehdad és mtsai, 2016), amely szűkítheti az optimális összefoglalók elkészítési lehetőségeit.

A tanulmányunk felépítése a következő: a 2. pontban röviden áttekintjük a vonatkozó szakirodalmat, a 3. pontban röviden bemutatjuk a vizsgálathoz használt adathalmazt. A 4. pontban bemutatjuk a mondatszámok meghatározásához használt megközelítéseket, az 5. pontban átvesszük milyen módszereket alkalmaztunk a vektorizálás során, végül a 6. fejezetben bemutatjuk a kapott eredményeinket.

2. Szakirodalmi áttekintés

Az automatikus tartalmi összefoglalót készítő rendszereket csoportosítani lehet aszerint, hogy azok csak egyetlen dokumentumhoz készítenek összefoglalót (*single document summarization*), vagy ezt dokumentumok egy csoportjához kapcsolódóan végzik el (*multi-document summarization*). A kivonat létrehozásának módja szerint pedig megkülönböztetjük a már említett absztraktív és extraktív metódusokat alkalmazó rendszereket, illetve felügyelt- és felügyelet nélküli gépi tanulást alkalmazókat is.

Extraktív kivonatokra tekintettel (Qiang és mtsai, 2016) például a szövegbányászatban (*text mining*) korábban sikerrel alkalmazott szekvencia feltáró módszerekkel (*sequential pattern mining*) készített több dokumentumra vonatkozó összefoglalókat, (John és mtsai, 2017) pedig ugyanezt a feladatot látens szeman-

tikus analízis (LSA), valamint nem-negatív mátrix faktorizálás (NMF) segítségével oldotta meg.

Egy dokumentumra vonatkozó extraktív összefoglalókat a jogi doménben például (Anand és Wagh, 2019) készített FFNN (Feed-Forward Neural Network) alkalmazásával, míg például (Elaraby és Litman, 2022) az egyes dokumentumok absztraktív összefoglalásának elkészítésében ért el eredményeket, többek között a BART (Lewis és mtsai, 2019) nyelvmodell alkalmazásával[1].

A magyar esetében az elmúlt években szintén a neurális háló alapú megoldások vannak többségben, mind az extraktív, mind az absztraktív kivonatok készítése kapcsán (pl. Yang Zijian és mtsai (2020); Yang Zijian (2022))

A mondatszám, vagy általánosan a kivonat hosszának meghatározása alapvetően három módszerrel közelíthető meg;

- fix érték megadása valamennyi kivonat esetében (*set*),
- kézzel kivonatolt dokumentumok, valamint a belőlük készített kivonatok mondatszámát alapul vévő módszer, például egyenes illesztésével az egyes dokumentumhosszokhoz tartozó kivonatok jellemző hosszának meghatározásához (*empirikus*), vagy
- automatikus meghatározással, például a dokumentumban található témák / témacsoportok szerint, mindegyikből n mondat kiválasztásával (*automatikus*).

Az egyik legkorábbi (modern értelemben vett) kísérlet a kivonatok optimális hosszának meghatározására (Goldstein-Stewart és mtsai, 1999) kísérletéhez kötődik. Ennek során a szerzők (inkább feltáró jelleggel) hírszövegek kézzel készített összefoglalóit vizsgálták meg, és vetették össze egy automatikus kivonatoló rendszer eredményeivel, amely statisztikai és nyelvi jegyeket egyaránt figyelembe vett. A kézzel készített összefoglalókkal történt összevetés során egyik legfontosabb megállapításuk az volt, hogy a kézi összefoglalók hossza jellemzően függetlenül alakult a dokumentum hosszától, továbbá, hogy az nagyjából 75-100 szó hosszúság között kulminált, amely támpontot adhat egy lehetséges fix hosszúság optimális értékéhez.

A neurális hálókon alapuló megoldások közül a Neusum (Zhou és mtsai, 2018) esetén (amely a mondatok pontozása és szelektálása alapján választja meg a kivonatba kerülő egységeket) ismert olyan irányú továbbfejlesztés (Nathan és mtsai, 2020) amelyben a modell a tényleges működés során képes kezelni az eltérő hosszúságú optimális összefoglalók kérdését. A modelltanítás során a szerzők figyelembe vették, hogy egy-egy mondatnak a kivonathoz csatolása meddig képes javítani a kivonat ROUGE pontszámát, és ezt az információt visszavezetve a hálózatba igyekeztek kezelni a nem minden esetben azonos hosszúságú optimális összefoglalók problémáját. Éppen ezért ez a módszer egyszerre tekinthető automatikus és empirikus megoldásnak is.

A kivonat hosszának automatikus meghatározására jó példa lehet az elemi diskurzusegységek (*Elementary Discourse Unit*) alapján (Keskes és mtsai, 2014)

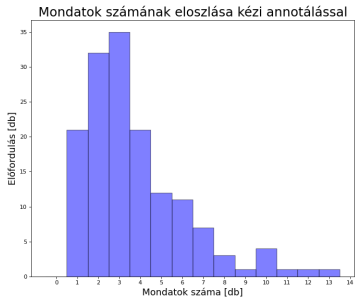
[1] A jogi doménben az elmúlt években (főleg neuronhálókkal) elért eredmények áttekintését részletesen lásd: (Shukla és mtsai, 2022).

történő szegmentálás, amely esetében az alapegységek ugyan nem mondatok, de a cél ugyancsak a szöveg tartalmának pontos reprezentációja változó hosszúságú kivonat elkészítésével.

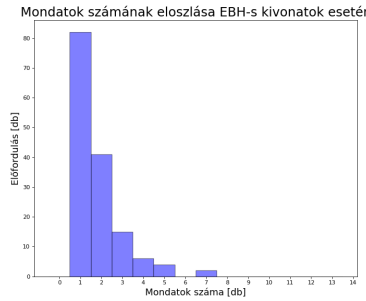
3. Adathalmaz

A vizsgálat elvégzésére 140 darab elvi bírósági határozat mellett 10 darab elvi bírósági döntést választottunk ki. Azért esett az ilyen típusú dokumentumokra a választásunk, mert ezek a dokumentumok valójában egy egyszerű bírósági határozathoz köthetők, tartalmilag lényegében megegyeznek ezekkel, azonban ki vannak bővítve egy pár mondatos kivonattal, amely kivonat az adott ügy jogi lényegét illetve relevanciáját tartalmazza. Így a dokumentum elejéről viszonylag könnyedén kinyerhetők ezek a kivonatok.

A mi célunk azonban az volt, hogy olyan kivonatot készítsünk, amely segít eldönteni, hogy az adott ügy releváns-e a felhasználó számára, vagy sem. Ezért felkértük a Wolters Kluwer Hungary Kft. munkatársait, hogy végezzenek el ilyen szemlélettel extraktív kivonatolást ezeken a dokumentumokon, megjelölve a kivonatként kiválasztott mondatokat. Nem tettünk semmilyen kikötést a mondatok számával kapcsolatban, azonban csak teljes mondatokat lehetett kijelölni. Magát a kivonatolást több ember is végezte. A kézi kivonatolás valamint az EBH-k elején található mondatok számainak eloszlását az 1. ábra mutatja be.



(a) Kézzel annotált



(b) EBH-kból származó

1. ábra. Mondatok számának eloszlása kézi illetve az EBH-k elején szereplő kivonatok esetében

Szembetűnő a kétféle megközelítés közti különbség az eloszlások alapján. Látható, hogy a kézzel annotált esetben sokkal több mondatot tartalmaztak a kivonatok, mint az EBH-k elejéről származó dokumentumok esetében. Az előbbieknél legnagyobb számossággal a 3 mondatos kivonatok bírtak, míg az utóbbiaknál az 1 mondatosak. Mindazonáltal mindkét esetben erős túlsúlyban voltak a rövid,

pár mondatos kivonatok, a kézzel annotált esetben egészen hosszú, 13 mondatos kivonatot is kaptunk, míg az EBH-k esetében a leghosszabb mindössze 7 mondatból állt.

4. Mondatszámok meghatározása

4.1. Mondatszegmentálás

A mondatokra történő bontást a *sentence-splitter*[2] nevű csomag segítségével végeztük el, amely egy többféle nyelven is használható heurisztikus mondatszegmentáló megoldás. Lehetővé teszi működés hangolását egy szólistával, amely olyan példákat tartalmaz, amely nem lehet egy mondat vége. A tapasztalataink alapján ez a megoldás jellemzően túlszegmentálta a mondatokat, így a jellemző hibákat, amelyeket nem lehetett kezelni szótár alapon, csak logikai alapon, reguláris kifejezések használatával javítottuk ki.

Három különböző megközelítést vizsgáltunk meg a mondatszámok meghatározására, melyeket alább mutatunk be részletesen.

4.2. Automatikus meghatározás

Automatikus meghatározást készítettünk oly módon, hogy a mondatokat vektorizáltuk, majd megvizsgáltuk K-Means algoritmust használva 1-10 klaszterre, hogy hogyan alakul a minták és a hozzájuk legközelebbi klaszterközéppont közötti távolságok négyzetösszege. Ezt követően (Satopaa és mtsai, 2011) munkában közöltek alapján határoztuk meg automatikusan az optimális klaszterszámot. Erre mutat be egy példát a 2. ábra.

Az automatikus meghatározás során normalizáljuk a kiszámított klaszterközéppontoktól való távolságokat (y), illetve a kipróbált klaszterszámokat is (esetünkben 1-10). Az optimális klaszterszámot a kezdeti- és végállapotot összekötő egyenes (referencia), valamint a normalizált értékek közti különbségből kaphatjuk meg oly módon, hogy ahol a kettő közti különbség (feketével jelölt az ábrán) a legnagyobb, az az automatikusan meghatározott töréspont (sárgával kiemelve).

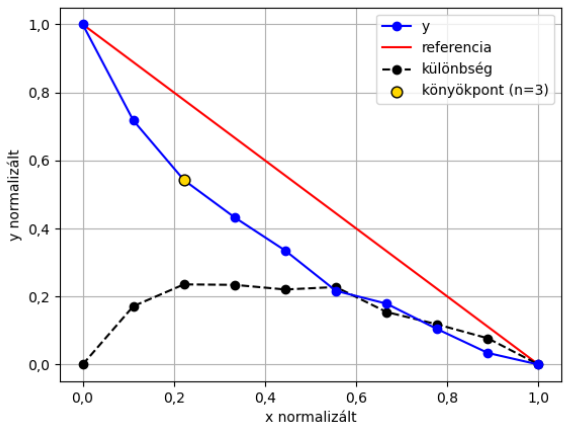
4.3. Egyenes illesztése

Egy másik lehetséges mondatszám-meghatározási módszer a kézzel annotált adat felhasználásával történő meghatározás. Megnéztük, hogy átlagosan hány mondat hosszúak az adott mondatszámú kivonathoz tartozó dokumentumok, majd ezen pontokra illesztettünk egy egyenest (lásd 3. ábra).

Látható, hogy általánosságban igaz volt, hogy az átlagos mondatszám növekedésével több mondat került a kivonatba. Ez némiképpen ellentétes a (Goldstein-Stewart és mtsai, 1999) által levont konklúzióval, miszerint ezek egymástól nagyrészt függetlenek.

A mondatszám meghatározásakor feltételként szabtuk még meg, hogy legalább 1, de maximum 10 mondatot adhat a módszer mondatszámosságként.

[2] https://pypi.org/project/sentence-splitter/



2. ábra. Klaszterszám automatikus meghatározása (Satopaa és mtsai, 2011)

4.4. Egyszerű beállítás

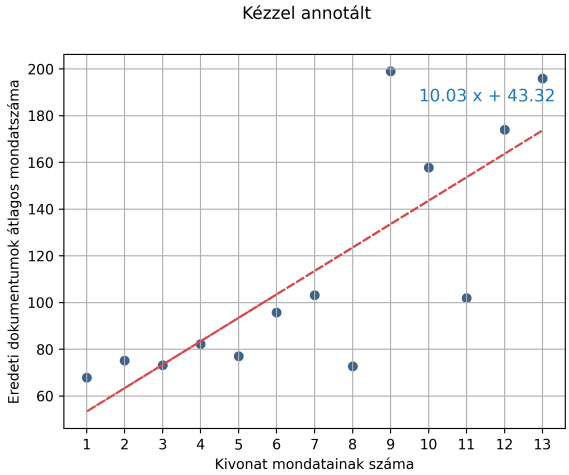
Megvizsgáltuk a legegyszerűbb megközelítést is, amely egy fix érték beállítását jelentette. Itt az emberek által készített kivonatok átlagos számához legközelebb eső értéket vizsgáltuk (4 mondat), valamint ennek közvetlen környezetét (3 illetve 5 mondat mindenhol).

5. Vektorizálás

Az automatikus módszer használatához, illetve a kivonatolás elvégzéséhez elengedhetetlen volt a szövegrészek vektorizálása. Ezt a mondatokra bontott szövegeknél a következő módszerekkel tettük meg:

- **Tf-Idf:** uni- és bigramokat tartottunk meg, kisbetűsítés, írásjel és stopszó szűrést végeztünk, valamint szótöveztünk a *hungarian-stemmer*[3] szótövezővel.
- **Doc2Vec:** 100 dimenziós modellt tanítottunk 5-ös szóablakkal, és megköveteltük a legalább 5-szöri előfordulást, minden más paramétert alapértelmezettként hagytunk (Le és Mikolov, 2014).
- **FastText:** A FastText hivatalos honlapon[4] elérhető magyar nyelvű modellt redukáltuk 100 dimenziósra, majd ennek segítségével a mondatokat vektorizáltuk (Bojanowski és mtsai, 2017; Mikolov és mtsai, 2018).
- **BERT:** A *sentence-transformers* (Reimers és Gurevych, 2019) csomagot alkalmaztuk a *huBERT* alap, nagybetűket is használó modelljével (Nemeskey, 2020, 2021; Devlin és mtsai, 2018).

[3] https://github.com/montana-knowledge-management/hungarian-stemmer
[4] https://fasttext.cc/docs/en/crawl-vectors.html



3. ábra. Átlagos mondatszámokra illesztett egyenes

6. Eredmények és következtetések

6.1. Eloszlások

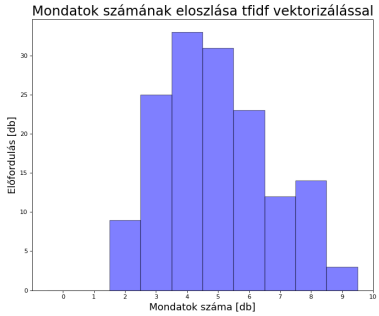
Első lépésként megvizsgáltuk az automatikus, K-Means klaszterezéssel történő mondatszám-meghatározás eredményeként kapott mondatszámok eloszlását. Az eredményeket az 1. táblázat, valamint a 4. ábra mutatja be.

1. táblázat. Különböző vektorreprezentációk hatása automatikus mondatszám-meghatározással

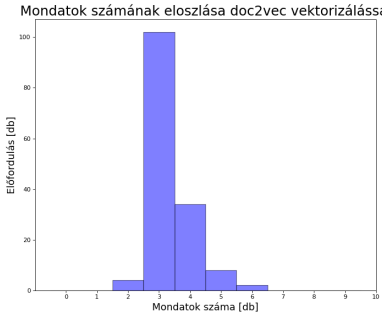
	Átlag	**Szórás**	**Min**	**Max**
kézi	3,72	2,41	1	13
ebh	1,78	1,15	1	7
tfidf	4,94	1,75	2	9
doc2vec	3,36	0,69	2	5
fasttext	3,82	0,82	3	7
bert	3,76	0,72	2	6

Az átlagértékek vizsgálatából kitűnik, hogy a kézi eredményekhez képest a legközelebb a BERT-tel kapott eredmények estek, másodikként a FastText, majd a Doc2Vec és végül a Tf-Idf.

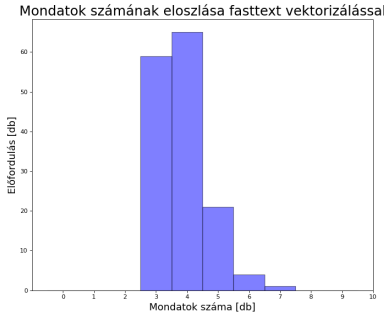
A 4. ábrán bemutatott eredmények alapján is kijelenthető, hogy jelentős különbségek adódtak a különböző vektorizálási megoldások szerint.



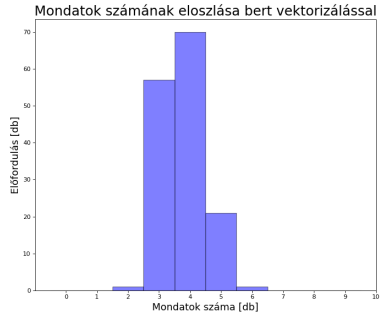
(a) Tf-Idf



(b) Doc2Vec



(c) FastText



(d) BERT

4. ábra. Mondatszám eloszlások automatikus mondatszám-meghatározás esetén

6.2. Átlagos eltérés

Összevetettük a különböző megközelítéseket a kézzel annotált mondatszámokhoz képest tapasztalt eltérések szerint is. Ehhez kiszámítottuk az átlagos négyzetes eltérés (ÁNE, Mean Squared Error, MSE) értéket a kézi annotálás eredményeihez képest. Egyenlet formában:

$$\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{1}$$

ahol x_i az i-edik dokumentumra meghatározott mondatszám, y_i pedig ugyanezen dokumentumra kézzel adott mondatok számát jelenti, N pedig a vizsgálathoz használt dokumentumok száma.

Az eredményeket a 2. táblázat tartalmazza.

2. táblázat. Átlagos négyzetes eltérések a kézzel kiválasztott kivonatok mondatszámához képest

	kézi	**fix 3**	**fix 4**	**fix 5**	**illesztett**	**automatikus**			
						tfidf	**doc2vec**	**fasttext**	**bert**
ÁNE (MSE)	0	6,32	5,88	7,44	9,64	9,38	6,4	6,22	5,76

Látható, hogy a fix beállítások közül a 4 mondatos bizonyult a legjobbnak átlagosan, míg az összes megközelítést figyelembe véve ez bizonyult a második legjobbnak. Ez nem meglepő, hiszen az 1. táblázat alapján a kézi kivonatolással kapott mondatátlaghoz (3,72) a 4 esik a legközelebb. Az abszolút legjobb eredményt a BERT esetében kaptuk az automatikus meghatározás segítségével. Kijelenthető, hogy a vektorizálás fajtája jelentős hatással volt az optimális mondatszám meghatározására. Megfigyelhető egy javuló tendencia a vektorizálási formák között, a legrosszabbnak a Tf-Idf bizonyult, majd sorban a Doc2Vec, FastText és BERT módszerek következtek, a legutóbbi az abszolút legjobb eredményt szolgáltatva. Ez a sorrend megegyezik azzal a sorrenddel, amelybe az egyes módszereket a képességük alapján soroltuk volna.

Érdekes eredmény, hogy a kézi eredményekre illesztett egyenes szolgáltatta a legnagyobb átlagos eltérést az összes vizsgált megközelítés közül. Ennek egyik oka, hogy bár mindegyik kivonatbeli mondatszámhoz a hozzájuk tartozó dokumentumok mondathosszának átlagát vettük, ezek a mondatszámosságok elég széles skálán szóródtak. Így az átlagokra illesztett egyenes is csak hozzávetőlegesen képes pontos eredményt szolgáltatni.

6.3. Mondatszámosság kivonat minőségére gyakorolt hatása

Megvizsgáltuk, hogy a kivonatok minőségében mennyire játszik jelentős szerepet a kiválasztott mondatok száma. A vizsgálathoz kiválasztottuk a két legjobban szereplő megközelítést az automatikus meghatározásból (BERT és FastText) valamint a legrosszabbul teljesítőt (Tf-Idf), a 4 mondatra fixált megoldást, valamint a legrosszabbul teljesítő illesztett megoldást.

A kivonatolást a következőképpen végeztük:

- kiszűrtük a dokumentumokból a kivonat szempontjából nem releváns részeket, ez a dokumentumok bevezető részét, valamint a dokumentum végén található perköltségekről, illetve a keltezést, és bírói neveket tartalmazó részeket jelentette,
- mondatokra bontottuk a szöveget,
- a mondatokat vektorizáltuk Tf-Idf, BERT és FastText módszerekkel,
- a TextRank (Mihalcea és Tarau, 2004) algoritmus segítségével sorba rendeztük a mondatokat,
- kivettünk a legrelevánsabb mondatok közül annyit, amennyit az adott mondatszámosság-meghatározó megközelítés adott.

Fontos megjegyezni, hogy az automatikus megközelítések esetében a vektorreprezentációk megegyeztek az optimális mondatszám-meghatározás valamint a kivonatolás közben.

Az így kapott kivonatokat összehasonlítottuk a kézi kivonatokkal a következő metrikák alapján: ROUGE-1, ROUGE-2, ROUGE-L F1 értékek, valamint kiszámítottuk a mondatok FastText és sentence BERT vektorreprezentánsait mind a gépi, mind a kézi kivonatok esetében is, melyek átlagai között mértünk cos távolságot (lásd 5. ábra). Ez utóbbi két metrikával igyekeztük a jelentésbeli közelségét

5. ábra. Kivonatok jelentéstartam szerinti összehasonlítása

mérni a kivonatoknak. A Rouge metrikák mérése előtt mindkét kivonatot kisbetűsítettük, szótöveztük, illetve kiszűrtük az írásjeleket. Ezeket az előfeldolgozási lépéseket nem végeztük el a FastText és a BERT cos távolságok számítása során.

A ROUGE metrikák olyan metrikák gyűjteményét jelentik, amelyek segítségével gépi fordítással készült szövegeket, vagy automatikus tartalmi összefoglalókat készítő szoftverek teljesítményét szokás számszerűsíteni Lin (2004). Segítségükkel meghatározható a kivonatolás kimenetének pontossága és fedése. Pontosság esetében arról kapunk visszajelzést, hogy a gép által az összefoglalóba beválogatott szavakból mennyi azok aránya, amelyek helyesen kerültek be az összefoglalóba, a fedés pedig azt mutatja meg, hogy a megtalálandó szavak közül mekkora arányban adta azokat vissza az algoritmus. Az egyes ROUGE metrikák e téren abban térnek el, hogy csak egy-egy szót tekintenek egységnek (ROUGE-1), két egymás melletti szó jelent egy egységet (ROUGE-2), vagy pedig az elvárt, és a kapott szövegek leghosszabb közös (nem feltétlenül megszakítás nélkül egymás után következő) részét (ROUGE-L).

Az eredményeket a 3. táblázat mutatja be, félkövérrel kiemelve a legjobb értékeket.

3. táblázat. Mondatszám-meghatározási módszer hatása a kivonat minőségére

Módszer	**Vektorforma**	**FastText cos**	**BERT cos**	**Rouge-1**	**Rouge-2**	**Rouge-L**
auto	BERT	0,963	0,9828	0,373	0,2008	0,259
fix 4	BERT	0,9643	0,9831	0,3825	0,2111	0,2669
illesztett	BERT	0,9552	0,9805	0,3291	0,1671	0,2262
auto	FastText	0,9689	0,9835	0,4083	0,2348	0,2873
fix 4	FastText	**0,97**	**0,9839**	**0,4142**	0,2387	0,2907
illesztett	FastText	0,9636	0,9805	0,3687	0,2034	0,259
auto	Tf-Idf	0,9563	0,98	0,4097	**0,2514**	**0,303**
fix 4	Tf-Idf	0,955	0,9795	0,4009	0,242	0,2967
illesztett	Tf-Idf	0,9481	0,9766	0,3719	0,22	0,2745

Az abszolút legjobb eredményt a cos távolságok esetén a fixen 4 mondatból álló FastText megoldás szolgáltatta, illetve ugyanez a megoldás bizonyult a legjobbnak a Rouge-1 metrika esetében is. A Rouge-2 és Rouge-L metrikáknál azonban az automatikus Tf-Idf megközelítés bizonyult a legjobbnak. Érdekesség, hogy a cos távolságok nagyon kis mértékben szórtak a BERT cos metrika esetében: a legjobb és legrosszabb értékek között a különbség mindössze 0,0073 volt, szemben a FastText cos távolságnál tapasztalt 0,0219-es értékkel. Ennek valószínű okát sikerült kideríteünk. Kézzel egy példamondaton leellenőriztük, hogy az általunk alkalmazott `sentence-transformer` megoldás ugyanazt a vektorreprezentációt szolgáltatja-e mint amikor kézzel kiválasztjuk a [CLS] token vektorreprezentációját, és azt tapasztaltuk, hogy ez a két vektor egymástól eltérő volt.

Mindhárom vektorizálási forma esetében igaz volt, hogy a különböző mondatszám-meghatározási módszerek közül az illesztett függvény segítségével meghatározott eset bizonyult a legrosszabbnak. A 2. táblázat alapján is ez a mondatkiválasztási metódus bizonyult a legrosszabbnak.

Az összes vektorforma esetében azonban az automatikus és a fix meghatározás viszonylag hasonló eredményt szolgáltatott, de a BERT és FastText esetekben a fix 4-es mondatkiválasztás esetén kaptuk a legjobb megoldásokat minden metrikánál, a Tf-Idf vektorforma esetében a legjobb eredményeket az automatikus meghatározás szolgáltatta. Ez meglepő eredmény a 2. táblázat tükrében, ugyanis a második legrosszabbnak az automatikus Tf-Idf bizonyult.

Az automatikus Tf-Idf kiemelkedő szereplésének feltehető oka, hogy a módszer átlagosan több, mint egy mondattal több mondatot választott a kivonatba (lásd 1. táblázat), mint a többi megoldás, ily módon nagyobb eséllyel tartalmazhatott olyan szövegrészeket, amelyek a kézi kivonatban is szerepeltek.

Az eredmények alapján tehát kijelenthető, hogy a mondatszám-meghatározás hatással van ugyan a kivonatok minőségére, azonban nem olyan mértékben, mint ami a 2. táblázatból következett volna. Kiszámítva az egyes vektorformák ese-

tén a különböző mondatszám-meghatározási módszerek esetén a legjobb és legrosszabb értékek közti különbségeket FastText cos metrika alapján, a következőket kapjuk: BERT: 0,0091, FastText: 0,0064, Tf-Idf: 0,0082. Megvizsgálva azt, hogy az egyes vektorformákkal kapott eredmények átlagai között mekkora a legjobb és legrosszabb közti eltérés, 0,014-et kaptunk. Az eredmények alapján arra a következtetésre juthatunk, hogy a mondatszám-meghatározás a vektorizálással összemérhető, azonban annál kisebb hatással bír a kivonat minőségére, ezért egy automatikus kivonat készítő projekt esetében fontos hangsúlyt fektetni az ideális hosszúság meghatározásra is.

7. Összefoglalás

Cikkünkben egy gyakorlati problémának, név szerint jogi dokumentumok extraktív kivonatolásának egy részfeladatát mutattuk be. A jellemzően több ezer szavas jogi határozatokból igyekeztünk kiválasztani azokat a mondatokat, amelyek a legjobban leírják az adott ügyet, ezzel segítve a jogi adatbázis felhasználóit, hogy könnyebben eldönthessék az adott ügy számukra releváns-e, vagy sem.

Extraktív, nem felügyelt kivonatolás esetében négy fő módon van befolyásunk a kivonat minőségére, a szöveg előszűrésével, a szöveg vektorreprezentációjával, a szövegrészek fontosság szerinti sorba rendezés megoldásával, valamint a kivonat szöveghosszának változtatásával. Jelen cikkben a kivonatok mondatszámának hatását vizsgáltuk a kivonatok minőségére.

Megmutattuk, hogy a mondatszám a vektorreprezentációs formával összemérhető, azonban annál kisebb hatással van a kivonat minőségére.

Hivatkozások

Anand, D., Wagh, R.: Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences (2019)

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the association for computational linguistics 5, 135–146 (2017)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

Elaraby, M., Litman, D.: Arglegalsumm: Improving abstractive summarization of legal documents with argument mining (2022), https://arxiv.org/abs/2209.01650

Goldstein-Stewart, J., Kantrowitz, M., Mittal, V., Carbonell, J.G.: Summarizing text documents: sentence selection and evaluation metrics. In: SIGIR '99 (1999)

John, A., Premjith, P., Wilscy, M.: Extractive multi-document summarization using population-based multicriteria optimization. Expert Systems with Applications 86, 385–397 (2017)

Keskes, I., Zitoune, F.B., Belguith, L.H.: Splitting arabic texts into elementary discourse units. ACM Transactions on Asian Language Information Processing 13(2) (jun 2014), https://doi.org/10.1145/2601401

Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196. PMLR (2014)

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019), https://arxiv.org/abs/1910.13461

Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)

Mehdad, Y., Stent, A., Thadani, K., Radev, D., Billawala, Y., Buchner, K.: Extractive summarization under strict length constraints. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). pp. 3089–3093. European Language Resources Association (ELRA), Portorož, Slovenia (May 2016), https://aclanthology.org/L16-1493

Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. pp. 404–411 (2004)

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)

Nathan, Z., Yijun, J., Yi, L.: Sentence-level extractive text summarization (March 2020), https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15790001.pdf, [Online; posted 20-March-2020]

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D.-értekezés, Eötvös Loránd University (2020)

Nemeskey, D.M.: Introducing `huBERT`. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). p. TBA. Szeged (2021)

Nenkova, A., McKeown, K.: Automatic summarization. Foundations and Trends® in Information Retrieval 5(2–3), 103–233 (2011), http://dx.doi.org/10.1561/1500000015

Qiang, J.P., Chen, P., Ding, W., Xie, F., Wu, X.: Multi-document summarization using closed patterns. Knowledge-Based Systems 99, 28–38 (2016)

Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084

Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops. pp. 166–171. IEEE (2011)

Schluter, N.: The limits of automatic summarisation according to rouge. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 41–45 (2017)

Shukla, A., Bhattacharya, P., Poddar, S., Mukherjee, R., Ghosh, K., Goyal, P., Ghosh, S.: Legal case document summarization: Extractive and abstractive methods and their evaluation (2022), https://arxiv.org/abs/2210.07544

Vági, R.: Szemantikai keresés és predictive coding a jogi munkában. In: Ződi, Zs. (szerk.) Jogi technológiák - digitális jogalkalmazás, pp. 141–156. Ludovika Egyetemi Kiadó (2022)

Yang Zijian, Gy.: Barterezzünk! messze, messze, messze a világtól, bart kísérleti modellek magyar nyelvre. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 15–29. Szegedi Tudományegyetem, Szeged (2022)

Yang Zijian, Gy., Perlaki, A., Laki, L.J.: Automatikus összefoglaló generálás magyar nyelvre bert modellel. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 343–353. Szegedi Tudományegyetem, Szeged (2020)

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences (2018), https://arxiv.org/abs/1807.02305