Absztraktív összefoglaló PreSumm módszerrel

Agócs Ádám[1], Yang Zijian Győző[1,2]

[1]MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport
1083 Budapest, Práter u. 50/a.
agadam98@gmail.com, yang.zijian.gyozo@itk.ppke.hu
[2]Nyelvtudományi Kutatóközpont
1068 Budapest, Benczúr u. 33.
yang.zijian.gyozo@nytud.hu

Kivonat Kutatásunk során egy szöveges összefoglaló szoftvert készítettünk magyar nyelvre, többnyelvű és magyar BERT alapú modellek felhasználásával. Alapvetően kétfajta szöveg összefoglalási módszert különböztetünk meg egymástól, extraktív és absztraktív. Az extraktív összefoglalók csak olyan szavakat, kifejezéseket tartalmaznak, melyek megtalálhatóak az eredeti, összegezni kívánt szövegben is. Ez a módszer az eredeti szövegben található, a legfontosabb szavak kiemelésével készíti az összefoglalót. Az absztraktív összefoglalás sokkal inkább hasonlít egy ember által összefoglalt szövegre, megjelenhetnek benne olyan szavak is, melyeket az eredeti szöveg nem tartalmaz. Kutatásunk során absztraktív modelleket tanítottunk magyar nyelvre. A modellekhez többnyelvű és magyar egynyelvű BERT modelleket használtunk. Létrehoztunk egy demó alkalmazást is, amelynek segítségével, valós időben is használhatjuk az összefoglaló rendszerünket. Jelen kutatásunkban a PreSumm kódot alapul véve készítettük el az absztraktív összefoglaló demónkat.
Kulcsszavak: absztraktív összefoglalás, BERT, huBERT, HILBERT, mBERT

1. Bevezetés

Az automatikus összegző alkalmazások használata jelentős idő- és költségmegtakarítást eredményezhet, ezért egyre nagyobb az igény az automatikusan működő alkalmazások iránt. Az automatikus szövegösszegzés különösen fontos, megoldatlan probléma a magyar nyelvre. Az automatikus szövegösszefoglaláshoz egy bemeneti szövegre van szükségünk, valamint egy összefoglaló rendszerre. A rendszer a bemeneti szövegből előállít egy összefoglalót, azaz oly módon csökkenti az eredeti szöveg hosszát, hogy a tartalma közben megmarad. Az összefoglalókat generáló technológiák figyelembe vesznek olyan változókat, mint a hossz, a stílus vagy a szintaxis. A hagyományos automatikus szövegösszegző módszerek a szöveg jellemzőinek logikai számszerűsítésére támaszkodnak, ideértve a kulcsszavak súlyozását, valamint a mondatok rangsorolását. Két különböző gépi összefoglaló módszer létezik: az extraktív és az absztraktív összegzés. Az absztraktív módszerrel elkészült összefoglaló tartalmazhat teljesen új szavakat is az eredeti

szöveghez képest úgy, hogy megtartja az eredeti szöveg jelentését. Az absztraktív módszerek általában bonyolultabbak, mivel a gépnek elemeznie kell a szöveget és a legfontosabb információkat, majd meg kell tanulnia a vonatkozó fogalmakat, és összefoglalót kell készítenie. Az extraktív módszerrel történő összefoglaló készítésekor az elkészült összefoglaló nem tartalmaz új szavakat az eredeti, összegezni kívánt szöveghez képest. Az eredeti szövegben található szavakat, mondatokat rangsorolja, és innen ragadja ki az összefoglalóhoz szükséges szövegrészeket. Az extraktív összefoglaló technikák közé tartozik a mondatok és kifejezések fontossági sorrendben való rangsorolása, valamint a dokumentum legfontosabb alkotóelemeinek kiválasztása az összefoglaló elkészítéséhez. Manapság a seq2seq neurális architektúra a legkiemelkedőbb, ahol egy neurális hálózat a bemeneti szekvenciákat a kimeneti szekvenciákhoz rendeli, ezen belül a transformer modellek érik el a legjobb eredményeket, melyek teljesen új utakat nyitottak az NLP feladatokban.

Kutatásunkban a neurális absztraktív összefoglalási módszert vizsgáljuk.

2. Kapcsolódó irodalom

Az elmúlt évek folyamán sokféle módszerrel közelítették a problémát. A Refresh (Narayan és mtsai, 2018) egy ROUGE (Lin, 2004) metrikán alapuló módszer, amelyet a mondatok rangsorolására használnak a megerősítéses tanulási módszer segítségével.

A Latent (Zhang és mtsai, 2018) a disztribúciós szemantika egyik technikája, amely elemzi a dokumentumok és a bennük található kifejezések közötti kapcsolatokat, feltételezi, hogy a közeli jelentésű szavak hasonló szövegdarabokban fordulnak elő, célja a kulcsszavak legpontosabb követése helyett az emberi munkával készült absztraktokhoz való minél közelebbi hasonlóság elérése volt.

A Sumo (Liu és mtsai, 2019) olyan módszert alkalmaz, amely a dokumentumból kinyerhető többgyökerű függőségi fa-struktúrákra épül, és az összefoglaló lehetséges formájának előbecslésén alapszik. A NeuSum (Zhou és mtsai, 2018) a mondatok pontozásával és szelektálásával közelíti meg a problémát.

A PTgen (See és mtsai, 2017) eszköz mutatókat (pointereket) generál a szavak azonosítására a forrásszövegben, majd egy lefedettségi mechanizmus használatával tartja meg az összefoglalóban felhasznált szavakat. A Deep Communicating Agent (Celikyilmaz és mtsai, 2018) olyan ágens alapú megközelítés, ahol az ágensek együtt reprezentálják a feldolgozandó dokumentumot és ennek dekódolásához kapcsolódik egy hierarchia figyelő ágens. Ezek a kódolók egyetlen dekóderhez vannak csatlakoztatva, kiképezve a végpontokat a megerősítéses tanulás segítségével, hogy fókuszált és koherens összefoglalót készítsenek. A Deep Reinforced Modell (Paulus és mtsai, 2018) olyan belső mechanizmust használ, amely különkülön figyeli a bemenetet és a folyamatosan generált kimenetet, valamint egy új képzési módszert, amely ötvözi a szabványos felügyelt szóbecslést és a megerősítő tanulást.

A BottomUp (Gehrmann és mtsai, 2018) megközelítés egy tartalomválasztót használ, az eredeti szövegben választja ki azokat a mondatokat, szavakat, melyeket tartalmazhatja a végleges összefoglaló.

A neurális hálóval végzett absztraktív összefoglalás a problémát egy szekvenciából egy másik szekvenciává való transzformálásként (sequence-to-sequence: seq2seq) közelíti meg. A feladathoz egy úgynevezett enkóder-dekóder architektúra szükséges. Az enkóder a változó hosszúságú forrás dokumentum tokenjeiből egy vektor reprezentációt készít, majd a dekóder az enkóder által készített vektor reprezentációjának segítségével, tokenről tokenre állít elő egy új szöveget.

A PreSumm (Liu és Lapata, 2019) eszköz számított a legmodernebbnek 2019-ben. Az extraktív és absztraktív összefoglaló modellek tanítását egy előre tanított BERT-modellre alapozza. Egy BERT-modell előtanításához rengeteg adatra és számítási kapacitásra van szükség. Szerencsére alkalmazhattuk a PreSumm eszközt, mivel az utóbbi időben több BERT modellt hoztak létre a magyar nyelv számára, valamint használhatjuk a többnyelvű BERT[1] modellt, amely tartalmaz magyar nyelvi információt is. Magyar nyelvre Nemeskey (2020a) hozott létre BERT modelleket, ezeket felhasználtuk kutatásainkhoz.

Az elmúlt időszakban több magyar nyelvű modellt tanítottak sikeresen[2]: ELECTRA, RoBERTa, ALBERT, BART és BERT large (Feldmann és mtsai, 2021).

Az elmúlt időszakban az autó regresszív módszerek érték el a legjobb eredményeket az összefoglalás területén. Ezen módszerek a Transofmer modell (Vaswani és mtsai, 2017) dekóderére támaszkodnak, és egy figyelem-maszkot használnak a teljes mondat tetején, így a modell csak az aktuális szöveg előtti tokeneket látja. Ez a módszer magasabb eredményeket ért el számos szöveggenerálási feladatnál (Yang és mtsai, 2019).

A BART (Lewis és mtsai, 2020) modell egy sequene-to-sequence modell, amely az enkóder oldalán maszkolás segítségével zajosítja a szövegeket, ebből megtanulja rekonstruálni az eredeti szöveget, ehhez kapcsolódik egy dekóder, ami a szöveggenerálásért felelős. Gyakorlatilag egy BERT Vaswani és mtsai (2017) jellegű modell összekapcsolása egy GPT (Radford és Narasimhan, 2018) jellegű modellel. Ez a modell rendkívül hatékony a szövegösszefoglaló feladatok finomhangolásához.

A T5 (Raffel és mtsai, 2020) szintén egy enkóder-dekóder modell, mely felügyelt és felügyelet nélküli feladatokon egyaránt lett tanítva, minden feladat szövegből-szöveg formátumú. Egy modellel tanul meg osztályozni, szöveget összegezni és gépi fordítani. Egy prefix segítségével különíti el a különböző feladatokat.

A PEGASUS (Zhang és mtsai, 2020) egy nyelvi modell, a fontos mondatokat eltávolítják/elfedik egy bemeneti dokumentumból, majd egy kimeneti sorozatként generálják ki őket a fennmaradó mondatokból, hasonlóan az extraktív összefoglalóhoz, a legjobb eredményeket éri el az absztraktív összefoglalás területén.

Magyar nyelven az OpinHu rendszer rendelkezik összefoglaló funkcióval (Miháltz, 2010). A rendszer kulcsszavakat és szövegkontextust használ az információkinyerésre. Lengyelné Molnár Tünde (Molnár Lengyelné, 2010) a kutatási kivonatok automatikus generálásának lehetőségeit és korlátait vizsgálta.

[1] https://github.com/google-research/bert/blob/master/multilingual.md
[2] https://hilanco.github.io/

A PreSumm (Liu és Lapata, 2019) eszköz segítségével Yang és társa megépítette az első magyar nyelvű extraktív összefoglaló eszközt (Yang és mtsai, 2020). Ebben a cikkben bemutatunk egy absztraktív összefoglaló eszközt, melyet a PreSumm rendszer felhasználásával hoztunk létre.

Jelen cikkünk első sorban a PreSumm eszközre koncentrál, ami része a TransformerSum-nak [3], de ebbe sajnos nem integrálható a modell, amit mi tanítottunk az eredeti PreSumm-on. Azonban végeztünk kísérleteket a TransformerSum eszközzel is.

3. A BERT modell

Kísérleteink során különböző típusú BERT modelleket használtunk fel az összefoglalók elkészítéséhez.

A BERT (Bidirectional Encoder Representation from Transformer) egy többrétegű, kétirányú Transformer enkódoló (Vaswani és mtsai, 2017). A BERT modell két nyelvi feladatra lett tanítva: maszkolás és következő mondat előrejelzés. A maszkolás során a korpuszban lévő szavak 15%-a véletlenszerűen elmaszkolásra kerül, majd a rendszer ezeket az elmaszkolt szavakat próbálja kitalálni. A következő mondat előrejelzésnél a modell kap két mondatot, az a feladat, hogy megmondja, a kapott két mondat az eredeti szövegben egymás mellett helyezkedik el, vagy csak két, a szövegből véletlenszerűen kiválasztott mondatról van szó. A szótár méretének korlátozásához, és az ismeretlen szavak problémájának kezeléséhez a WordPiece (Schuster és Nakajima, 2012) tokenizáló került felhasználásra.

A BERT egyik nagy előnye, hogy a modelleket nem csak angolul tanították. A Google létrehozott két többnyelvű modellt[4] is, egyik a kisbetűs másik, a nem kisbetűs. A modelleket a 104 legnagyobb Wikipédiával rendelkező nyelven tanították. Ezen nyelvek Wikipédia mérete nagyban különbözik egymástól, az adatok közel 20%-át az angol Wikipédia teszi ki, ezért normalizálással kontrollálták a mintavételezést a probléma elkerülése érdekében. Ezután minden nyelvet tokenizálásnak vetettek alá, ami négy lépésből állt: kisbetűsítés, ékezetek eltávolítása, írásjelek leválasztása, whitespacek kezelése. A nem kisbetűsített modell is ezeken a lépéseken esett át, a WordPiece szótárral kezelik a nem kisbetűs és ékezetes szavakat. A WordPiece tokenizálás és szótár kezeli a kisbetűs és ismeretlen szavakat. A magyar nyelv is része ennek a modellnek.

Az első magyar BERT modellt Nemeskey (2020b) tette közzé, mely a huBERT[5] nevet kapta. Három huBERT modell született:

- huBERT: Magyar Webkorpusz 2.0-n[6] tanított BERT base modell
- huBERT Wikipedia cased: Magyar Wikipédián tanított nem kisbetűsített BERT base modell

[3] https://github.com/HHousen/TransformerSum
[4] https://github.com/google-research/bert/blob/master/multilingual.md
[5] https://hlt.bme.hu/en/resources/hubert
[6] https://hlt.bme.hu/en/resources/webcorpus2

– huBERT Wikipedia lowercased: Magyar Wikipédián tanított kisbetűsített BERT base modell

Jelenleg a huBERT modellek érik el a legjobb eredményeket a névelem felismerésében, valamint a főnévi csoportok felismerésében (Nemeskey, 2020a).

4. Felhasznált korpuszok és modellek

A finomhangoláshoz használt korpuszok felépítéséhez 4 különböző forrást használtunk fel: HVG[7], index.hu[8], nol.hu[9] (NOL) és a magyar MARCELL korpusz (Váradi és mtsai, 2020). Az 1. táblázat a korpuszok fő jellemzőit mutatja be.

A HVG, az index.hu és a NOL esetében a napi online újságból vettük a cikkek törzsét, valamint az összefoglalókat reprezentáló leadeket. A HVG és az index.hu esetében két korpuszt építettünk belőlük. Az első változatban csak a HVG dokumentumokat használtuk. A második változatban (H+I korpusz) egyesítettük a HVG és az index.hu cikkeit. A MARCELL esetében jogi dokumentumokat használtunk forrásként, mely mindegyikéhez tartozik egy rövid mondatos témaleírás, amelyet az összegzéshez alkalmaztunk.

	HVG	index.hu	H+I	MARCELL	NOL
Év	2012–2020	1999–2020	-	1991–2019	1999–2016
Dokumentumok	480.660	183.942	559.162	24.747	397.343
Token	129.833.741	104.640.902	159.131.373	28.112.090	168.789.330
Type	5.133.030	3.921.893	3.053.703	450.115	2.589.211
Átlagos token # - cikk	246,27	496,27	265,17	1124,82	384,52
Átlagos token # - lead	12,43	22,33	29,97	11,22	39,71
Átlagos mondat # - cikk	23,74	35,76	11,40	49,26	17,36
Átlagos mondat # - lead	1,46	2,23	1,57	1,00	1,86

1. táblázat. A korpuszok fő jellemzői.

A BERT modellnek van egy maximum 512 szóelem (subword) hosszú megkötése (a BERT tokenizálása után), ezért kutatásunkban csak az online napi cikkeket és a hozzájuk tartozó leadeket használtuk, mert a hetilap (HVG) cikkei sokkal hosszabbak. A MARCELL esetében az átlagos mondathossz 1124,82 szóelem, ami jóval több, mint az 512, de a medián csupán 340, ami elég rövid ehhez a feladathoz.

Első feladatként (kivétel a csak HVG korpuszt használó kísérlet és MARCELL korpuszt) különböző tisztítási folyamatokat végeztünk. A tisztítási és normalizálási szempontok a következők:

[7] https://hvg.hu
[8] https://index.hu
[9] http://nol.hu

– Eltávolításra kerültek a hosszú (500<token) dokumentumok a korpuszból.
– Eltávolításra kerültek a rövid (5>token) dokumentumok a korpuszból.
– Eltávolításra kerültek azon cikkek, melyek rövidebbek voltak, mint a hozzájuk tartozó lead (Például lásd: 5. táblázat).
– Eltávolításra kerültek az irreleváns cikk részletek, mind például a: "Kövessen minket Facebook-on", "Kattintson további részletekért", "Kvíz indítása" stb.
– Eltávolításra kerültek azok a dokumentumok, amelyek szkripteket tartalmaztak.

Absztraktív összefoglaló kísérleteink során 4 különböző típusú, előre betanított BERT modellt használtunk: huBERT, huBERT Wikipedia cased, HILBERT, BERT-Base-Multilingual-Cased.

A **huBERT** (Nemeskey, 2020a) a jelenlegi „state-of-the-art" magyar (nem kisbetűs) BERT base modell, amely a Webcorpus 2.0-án[10] (9 milliárd token, 110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej) tanított.

A **huBERT Wikipedia cased** (Nemeskey, 2020a) egy magyar BERT base modell, amely a magyar Wikipédián tanított (170 millió token, 110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej).

A **HILBERT** (Feldmann és mtsai, 2021) egy magyar nyelvű BERT large modell, amely az NYTK v1 korpuszon (3,7 milliárd token, 340 millió paraméter, 24 réteg, 1024 rejtett réteg méret, 16 figyelmi fej) tanított.

A **BERT-Base-Multilingual-Cased**[11] egy többnyelvű BERT base modell, a modell tanításához kiválasztották az első 104 nyelvet, amely a legnagyobb Wikipédiával rendelkezik (110 millió paraméter, 12 réteg, 768 rejtett réteg méret, 12 figyelmi fej).

5. Kísérletek

Az előre betanított nyelvi modellek segítségével finomhangoltuk a magyar nyelvű absztraktív összefoglaló modelljeinket. Kutatásunk első lépése az eredeti szöveg előzetes feldolgozása volt. A cikkeket és a hozzájuk tartozó leadeket, az e-magyar tokenizálóval[12], a quntoken (Mittelholcz, 2017) eszközével tokenizáltuk. Ezután a tokenizált szöveget JSON formátumba konvertáltuk az összefoglaló rendszer számára. A rendszer ezután beilleszt két speciális elemet, az első a szöveg elejét jelzi, a másik pedig a mondathatárokat. Az előfeldolgozás után különböző összefoglaló modelleket tanítottunk.

Munkánk során sokat kísérleteztünk a TransformerSum-mal, ami magába foglalja a PreSumm eszközt is. A TransformerSum egy olyan könyvtár, melynek segítségével képesek vagyunk tanítani, kiértékelni valamint használni különböző szöveg összegzésre használatos Transformer modelleket. A TransformerSum támogatja mind az extraktív, mind az absztraktív, valamint a hosszú mondatos (4.096 - 16.384 token) összefoglalók elkészítését. Ezt extraktív összefoglalók

[10] https://hlt.bme.hu/en/resources/webcorpus2
[11] https://github.com/google-research/bert/blob/master/multilingual.md
[12] https://e-magyar.hu

esetében a longformer, absztraktív összefoglaló esetében pedig a LongformerEncoderDecoder -rel oldja meg. A TransformerSum olyan modelleket is tartalmaz, amelyek korlátozott erőforrás eszközökön is futnak, miközben továbbra is nagy pontosságot biztosítanak. A modellek kiértékelése automatikusan történik a ROUGE segítségével. Rengeteg problémát okozott a TransformerSum integrálása, egyes rendszereinken a mai napig nem sikerült működésre bírni, más rendszerünkön sikeresen el tudtuk indítani, de számos problémába ütköztünk, rengeteg kompatibilitási, valamint függőségi probléma fedte fel magát, melyek egyes rendszereinken megoldatlanok. Továbbá kódszinten ki kellett egészíteni az implementációt, mivel a párhuzamosítás során ellentmondásos beállításokat észlelt a rendszer. Ilyen például az „amp_backend" beállítása „apex" értékre, valamint a „Trainer" függvény „strategy" paraméter beállítása. Végül, de nem utolsó sorban a saját adat használata esetén az adat struktúrájáról sem volt leírás, azt is hosszas utánajárás után sikerült kideríteni. A sok probléma ellenére az egyik rendszerünkön sikeresnek bizonyult a próbálkozásunk, a H+I korpuszunkon egy absztraktív modell tanítását sikerült elindítani. Sajnos azonban több mint 20 tanítás után sem sikerült a megfelelő hiperparaméter beállításokat megtalálni, amivel a modell konvergálna és értelmes összegzést tudna tanulni. Ezért ezen a területen további kísérletek szükségesek még.

Az absztraktív modellek tanításának másik vonala a PreSumm (Liu és Lapata, 2019) eszközzel[13] való kísérletek. A PreSumm módszerrel könnyebben lehet a tanítást elvégezni.

A 2. táblázatban láthatjuk a BERT base modellek és a BERT large modell módosított tanítási (finomhangolási) hiperparamétereit. Az összes többi hiperparaméter Liu és Lapata (2019) kísérleteinek alapértelmezett értékeire voltak beállítva.

	tanulási ráta (tr)	tr csökkentés	batch méret	hardver
BERT base	1e-03	0,1	20	4x GeForce RTX 2080 (12GB)
BERT large	5e-05	0,02	10	4x Tesla V100 (32GB)

2. táblázat. A BERT base és a BERT large módosított hiperparaméterei.

A kísérleteink során azt tapasztaltuk, hogy minél nagyobb a korpusz, annál több lépésre van szükség. Ennek megfelelően a következő tanítási lépéseket alkalmaztuk:

– Absztraktív összefoglalás:
  - HVG: 200.000
  - H+I: multi és huBERT: 600.000; HILBERT: 800.000
  - MARCELL: 50.000
  - NOL: 600.000

[13] https://github.com/nlpyang/PreSumm

Kísérletünk során végeztünk transzfer kísérletet is, ami azt jelenti, hogy a multi-BERT modellt először az eredeti CNN/Daily Mail korpuszon[14] finomhangoltuk 200.000 lépésszámmal, majd azt tovább finomhangoltuk a magyar korpuszokon, remélve, hogy az angol finomhangolásból is tanul fontos információkat.

6. Eredmények és kiértékelés

Az eredmények kiértékeléséhez a ROUGE (Lin, 2004) metódus került felhasználásra. A ROUGE (Recall-Oriented Under-study for Gisting Evolution) egy fedés (recall) alapú módszer, mely a gépi fordításban használt BLEU metrikán alapszik. Maga a ROUGE számos módszert tartalmaz, ezek közül a ROUGE-1, a ROUGE-2 valamint a ROUGE-L módszereket használtuk a mérésekhez. A ROUGE-1 egy uni-gram, míg a ROUGE-2 egy bigram fedést számító algoritmus. A ROUGE-L a bekezdések és mondatok szintjén vizsgálja a leghosszabb közös szószekvenciát.

A 3. és 4. táblázatban láthatóak az absztraktív modellek ROUGE eredményei. Mivel a HILBERT modell hatalmas erőforrásokat igényel, csak a H+I kísérletében használtuk, és ebben a feladatban nem a huBERT wikit használtuk, mert a huBERT tartalmazza a wikit. A MARCELL esetében, a generált kimenetnek az első mondatát vettük csak, mivel a referencia is csak egy mondatból áll.

		ROUGE-1	ROUGE-2	ROUGE-L
MARCELL (1. mondat)	multi	87,37	77,38	84,97
	huBERT wiki	89,37	79,91	86,14
	huBERT	**89,64**	**80,29**	**86,46**
HVG	multi	47,02	19,72	39,29
	huBERT wiki	49,49	21,62	41,46
	huBERT	**51,47**	**23,27**	**43,82**
H+I	multi (600k)	51,85	23,22	43,45
	multi transfer (650k)	51,61	22,25	42,85
	huBERT (450k)	**57,07**	**26,97**	**48,28**
	HILBERT (800k)	44,98	14,22	37,06
NOL	multi (600k)	43,41	17,24	35,70
	multi transfer (600k)	43,42	16,26	35,09
	huBERT (750k)	**51,18**	**22,61**	**43,03**
CNN/Daily Mail – multi		60,32	25,79	56,91

3. táblázat. ROUGE fedési eredmények.

Az eredményben az látható, hogy a magyar huBERT modellek minden esetben felülmúlják fedés értékben a multi-BERT és HILBERT eredményét. A 3. táblázatban a fedési (recall) értékeket láthatjuk, mivel a modell több mondatot

[14] https://github.com/abisee/cnn-dailymail

		ROUGE-1	ROUGE-2	ROUGE-L
MARCELL (1. mondat)	multi	72,99	65,38	71,53
	huBERT wiki	74,23	66,56	72,90
	huBERT	**75,85**	**68,35**	**74,61**
HVG	multi	**26,92**	**10,63**	**22,26**
	huBERT wiki	21,50	8,67	17,81
	huBERT	21,69	9,09	18,27
H+I	multi (600k)	**28,34**	**12,40**	**23,45**
	multi transfer (650k)	27,81	11,71	22,81
	huBERT (450k)	22,42	10,24	18,73
	HILBERT (800k)	17,36	5,41	14,14
NOL	multi (600k)	**30,56**	**11,57**	**24,99**
	multi transfer (600k)	30,34	10,83	24,42
	huBERT (750k)	26,53	11,08	22,19
CNN/Daily Mail – multi		25,76	10,91	24,37
CNN/Daily Mail – BERT		41,72	19,39	38,76

4. táblázat. ROUGE F-mérték eredmények.

is generál, és az esetek nagy részében a generált mondatok száma meghaladja a referencia mondatok számát, ezért a pontosság (precision) alacsony.

A H+I és NOL korpuszok esetében zárójelben láthatjuk a legjobb eredményt elérő lépésszámokat. A lépésszámok alapján a huBERT 450.000 lépésnél érte el a legjobb eredményeket, sokkal korábban, mint a többi modell. A HILBERT esetében nem értük el az elméleti optimumot, mert a ROUGE értékek folyamatosan nőttek. A 3. táblázatból látható, hogy a HILBERT teljesítménye sokkal alacsonyabb, mint a többi modellé, mivel a HILBERT large méretű, kétszer annyi paraméterrel rendelkezik, mint a BERT base, a modell robusztusabb és a finomhangolás nehezebb. 47 sikertelen kísérlet után találtuk meg azt a hiperparaméterek halmazát (Lásd: 2. táblázat), amelyekkel a modell konvergált. Úgy véljük, hogy a HILBERT magasabb eredményeket érhet el, de további kísérletekre van szükségünk ahhoz, hogy megtaláljuk a legjobb hiperparamétereket a legmagasabb eredmény eléréséhez.

A 4. táblázatban láthatóak az F-mérték eredmények. A nemzetközi irodalomban is az F-mérték a mérvadó. A PreSumm eszköz jellege miatt, hogy több és sokkal hosszabb mondatot is generál, mint a referencia, a fedés mértékek inkább a relevánsabbak. Példának a H+I korpuszban:

– Eredeti leadek méretei: átlag token szám: 29,97 (Lásd: 1. táblázat).
– PreSumm összefoglalók méretei: átlag token szám: 104,61.

Azonban fontos megvizsgálni az F-mértékeket is. Az érdekesség, ami látható a 4. táblázatban, hogy a multi-BERT modellek javítják a pontosságot (precision), bár fedésben gyengébbek, F-mértékben erősebbek, ami azt jelenti, hogy tömörebben tudnak releváns összefoglalókat generálni (kivétel a MARCELL kísérletek). Továbbá MARCELL esetében érdekes az F-mérték, mivel ott pontosan egy

mondatot veszünk figyelembe. Azonban itt a huBERT minden esetben a legjobb eredményt nyújtja.

A transzfer kísérletek is láthatóak az eredményekben. Az látható, hogy a transzfer tanításaink nem javították az eredményt egy esetben sem (egy esetet leszámítva, a NOL ROUGE-1 fedés esetében, de csak 1 századnyi a különbség).

Láthatunk néhány példát a 5. és a 6. táblázatban, amelyeket absztraktív összefoglaló modelljeink generáltak. A példákat elemezve észrevehetjük modelljeink néhány közös vonását. Ha a cikk hosszú (Lásd: az 5. táblázat), a modellünk kivonja a kifejezéseket az eredeti cikkből, majd egyesíti őket új mondatok létrehozásával. Hasonló az extraktív modellekhez, a különbség az, hogy az extraktív modellek teljes mondatokat választanak a cikkből, és rangsorolás után adják vissza a felhasználónak. Általában az absztraktív modellek által előállított mondatok nyelvtanilag többnyire helyesek. Minden modell több mondatot generál, de a végére "elfogy", és mondatfoszlányokat hagyhatnak (Lásd: a 5. táblázat).

Ha a cikk rövid (Lásd: az 6. táblázat), a modellek megmutatják valódi absztraktív tulajdonságukat, vagyis olyan részeket generálnak, amelyeket az eredeti cikk nem tartalmazott. Ebben az esetben túl kevés információ található az eredeti cikkben, így a kimenet többet „hallucinál” és a teljesítménye alacsonyabb.

A példákat vizsgálva láthatjuk a ROUGE metódus hátrányait, valamint a leadek felhasználásának problémáját. A ROUGE mutató csak azt mutatja, hogy a generált kimenet mennyire hasonlít a leadhez. A lead szerepe azonban gyakran az, hogy felhívja magára a figyelmet, vagy nagyon tömören csak a lényeget írja le, és nem az, hogy összegezze a cikk szövegét. A 1. példában (Lásd: az 5. táblázatot) a cikk egy teljes esetet leír, azonban a lead egy nagyon szűkszavú szöveg, míg a modellek sokkal részletesebben adják vissza a szöveget. Ez az egyik oka annak, hogy az eredményekben (Lásd a 3. táblázatot) csak körülbelül 50 %-os fedési eredményt láthatunk. A példában azt is láthatjuk, hogy a multi-BERT rövidebb összefoglalókat generál a huBERT-hez képest. A nagyon hosszú szövegek generálásának másik hibája az, hogy az összefoglaló olyan hosszú, mint a cikk maga, ami elveszíti összefoglaló jellegét. De azért az esetek többségében hosszabbak a cikkek, példának a H+I korpuszon:

– Eredeti cikkek méretei: átlag token szám: 496,27 (Lásd: 1. táblázat).
– PreSumm összefoglalók méretei: átlag token szám: 104,61.

7. Demó alkalmazás

A PreSumm eszközből kiindulva elkészítettünk egy demó alkalmazást, melyet a Docker program segítségével hoztunk létre. A Docker[15] segítségével különböző konténereket tudunk kezelni, melyek egymástól elkülönítve különböző alkalmazásokat, könyvtárakat és eszközöket kötnek össze. A konténerek imagefile-okból jönnek létre. Minden konténert egy operációs rendszer kernel működtet, így kevesebb a rendszerigénye, mint a virtuális gépeknek.

[15] https://www.docker.com

Cikk
Kedd délután a rendőrség megerősítette az Index nek , hogy reggel elfogtak a rendőrök egy férfit Budapesten , aki lőfegyverrel a kezében álldogált a Nagyvárad téren a 24-es villamos megállójában . A lapnak több szemtanú is beszámolt az esetről , a rendőrség pedig azt közölte , hogy kedd reggel 7 óra 12 perckor bejelentést kaptak arról , hogy a Nagyvárad tér 1. szám alatti villamosmegállóban egy fegyver van egy ember kezében . Lövés nem dördült , és senki nem sérült meg , a járőrök perceken belül a helyszínre érkeztek , és intézkedés alá is vontak egy 49 éves budapesti férfit . P. Lászlónál egy gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatos szabálysértés miatt előállították , és eljárást kezdeményeztek vele szemben .
Lead
Lövés nem dördült , a férfit elfogták .
huBERT web
Garázdaság és lőfegyverrel kapcsolatos szabálysértés miatt előállították P. Lászlót , akinél gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatban szabálysértés miatt eljárást kezdeményeztek vele szemben . Lövés nem dördült , senki nem sérült meg , de a rendőrök perceken belül elfogták a férfit . P. Lászlónál egy gáz-riasztó fegyvert is találtak . Garázdaság miatt előállították , és eljárást kezdeményeztek ellene . Garázdaság és lő fegyverrel kapcsolatos szabálysértése miatt előállították . A férfi ellen eljárást kezdeményeztek , és eljárás indult ellene . Videó
HILBERT
A férfi fegyverrel a kezében sétált a 24-es villamosmegállóban , a rendőrök perceken belül elfogták , és kezdeményezik előzetes letartóztatásukat . A rendőrség eljárást indított vele szemben , és eljárást indítottak vele szemben . A férfi ellen eljárás indult . A rendőrség szerint egy gáz-riasztó fegyvert is találtak a rendőrök , és elrendelték az előzetes letartóztatását . Az MTI úgy tudja , hogy a férfi egy gáz-riasztó fegyverrel is felszerelkezett a 24-es villamos megállójában . Videó is van a helyszínről , a rendőrség a hvg.hu-nak arról számolt be , hogy kedd reggel fogtak el egy férfit , aki lőszerrel a rendőrök elől menekülve próbált meg egy embert a 24-es villamoson . hu hu
multi-BERT
Kedd reggel elfogták a rendőrök azt a férfit , aki lőfegyverrel a kezében álldogált a Nagyvárad téren a 24-es villamos megállójában . Lövés nem dördült , és senki nem sérült meg , a járőrök perceken belül a helyszínre érkeztek , és intézkedés alá is vontak egy 49 éves budapesti férfit . P. Lászlónál egy gáz-riasztó fegyvert találtak , ezért garázdaság és lőfegyverrel kapcsolatos szabálysértés

5. táblázat. 1. példa absztraktív modellre (H+I korpusz).

Szerver-kliens architektúrán alapul a demó, azaz a szerver végzi a számítást, amelynek egy kliensen keresztül adhatjuk át az összegezni kívánt szöveget. Az alkalmazásban, a Flask keretrendszer[16], valamint az uWSGI program fontos szerepet játszik. Az uWSGI[17] egy szoftveralkalmazás, amelynek célja full stack alkalmazások fejlesztése a tárhelyszolgáltatásokra. Az uwsgi egy natív bináris protokoll, amelyet az uWSGI használ más szerverekkel való kommunikációhoz. A Flask egy webes keretrendszer (Python modul), amely lehetővé teszi a we-

[16] https://flask.palletsprojects.com
[17] https://uwsgi-docs.readthedocs.io

Cikk
A Daily Mail Twitter-oldalán közzétett videó szerint most épp nem történt nagy baj , de ez könnyen történhetett volna másképp is .
Lead
Valószínűleg nem gondolták át igazán ezek a fiatalok mit tesznek .
huBERT web
A Daily Mail Twitter-oldalára került fel egy videó arról , ahogy az egyik brit autópályán egy motorcsónaknaknak csapódott . A videón jól látható , hogy a sofőrnek nem sok esélye van a balesetre , de a videón jól látszik , hogy nem is kell aggódnia . És nem is ez volt az első ilyen eset . A videóban rögzítették , ahogy a britek egy hatalmasat ugranak a fejük előtt . Az alábbi videón látható , hogyan történt a dolog , és hogyan lett volna ez a videóból . A videón látható .
HILBERT
A Magyar Kétfarkú Kutya Párt szerint most már nem is történt nagy baj , de ez így is elég lett volna . nem ez volt az első eset , hogy nem lett komoly baj . " Ez történt volna " - reagált a párt . Videón az is látszik , hogy a párt szerint nem történt baj . Igaz , ez még így is így lett volna , ha nem is lett volna baj , az eredmény nem lett volna ennyire egyszerű .
multi-BERT
A Daily Mail videója szerint nem történt nagy baj . A videó könnyen előfordulhatott volna , hogy egy hatalmas bajba került . Nem kellett volna . Videó . Szerencsére nem történt baj , de ez könnyen történhetett volna másképp is . De nem . Videón . Mutatjuk . Szerencsésen volt . Videót . És . Munkatársunk . Sőt . Mutatja !Videó !Szavazzon !! A hvg.hu hu . A

6. táblázat. 2. példa absztraktív modellre (H+I korpusz).

bes alkalmazások egyszerű fejlesztését. Kicsi és könnyen bővíthető. Az elkészült demóval képesek vagyunk valós időben is összefoglalókat készíteni. A demó és további példák a projekt oldalunkon[18] érhetőek el.

8. Összegzés

Különböző absztraktív szövegösszefoglaló eszközöket hoztunk létre a magyar nyelvre. Az összefoglaló modellek elkészítéséhez különböző BERT alapú modelleket használtunk. Az absztraktív modellekhez az előre betanított többnyelvű BERT modellt, valamint a magyar egynyelvű huBERT base, valamint a HILBERT large modelleket használtuk. Továbbá végeztünk transzfer tanítást is. A BERT alapú modellek finomhangolásához, az összefoglalók elkészítéséhez, a PreSumm eszközt használtuk.

Az eredmények azt mutatják, hogy az egynyelvű magyar modellek minden esetben felülmúlták a többnyelvű modellt fedés szempontjából, azonban ha az F-mértéket nézzük, a multi modellek teljesítenek jobban, ami azt jelenti, hogy tömörebben tudnak generálni, mint a magyar modellek. A transzfer tanítással azonban nem tudtunk elérni eredményjavulást.

[18] http://nlpg.itk.ppke.hu/projects/summarize

Hivatkozások

Celikyilmaz, A., Bosselut, A., He, X., Choi, Y.: Deep communicating agents for abstractive summarization. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1662–1675. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)

Feldmann, A., Hajdu, R., Indig, B., Sass, B., Makrai, M., Mittelholcz, I., Halász, D., Yang, Z.G., Váradi, T.: Hilbert, magyar nyelvű bert-large modell tanítása felhő környezetben. XVII. Magyar Számítógépes Nyelvészeti Konferencia pp. 29–36 (2021)

Gehrmann, S., Deng, Y., Rush, A.: Bottom-up abstractive summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4098–4109. Association for Computational Linguistics, Brussels, Belgium (2018)

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. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics (2020)

Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)

Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 3730–3740. Association for Computational Linguistics, Hong Kong, China (2019)

Liu, Y., Titov, I., Lapata, M.: Single document summarization as tree induction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1745–1755. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

Miháltz, M.: Opinhu: online szövegek többnyelv véleményelemzése. VII. Magyar Számítógépes Nyelvészeti Konferencia pp. 14–23 (2010)

Mittelholcz, I.: emtoken: Unicode-képes tokenizáló magyar nyelvre. XIII. Magyar Számítógépes Nyelvészeti Konferencia pp. 61–69 (2017)

Molnár Lengyelné, T.: Automatic abstract preparation. 10th International Conference On Information: Information Technology Role in Development pp. 550–561 (2010)

Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1747–1759. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)

Nemeskey, D.M.: Egy embert próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged (2020a)

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

Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada (2018)

Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research 21(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html

Schuster, M., Nakajima, K.: Japanese and korean voice search. In: ICASSP. pp. 5149–5152. IEEE (2012)

See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. Association for Computational Linguistics, Vancouver, Canada (Jul 2017)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (szerk.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)

Váradi, T., Koeva, S., Yamalov, M., Tadić, M., Sass, B., Nitoń, B., Ogrodniczuk, M., Pęzik, P., Barbu Mititelu, V., Ion, R., Irimia, E., Mitrofan, M., Păiș, V., Tufiș, D., Garabík, R., Krek, S., Repar, A., Rihtar, M., Brank, J.: The MARCELL legislative corpus. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3761–3768. European Language Resources Association, Marseille, France (May 2020)

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (szerk.) Advances in Neural Information Processing Systems. vol. 32, pp. 5753–5763. Curran Associates, Inc. (2019)

Yang, Z.G., Perlaki, A., Laki, L.J.: Automatikus összefoglaló generálás magyar nyelvre bert modellel. XVI. Magyar Számítógépes Nyelvészeti Konferencia pp. 343–354 (2020)

Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences forabstractive summarization. In: Thirty-seventh International Conference on Machine Learning (2020)

Zhang, X., Lapata, M., Wei, F., Zhou, M.: Neural latent extractive document summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 779–784. Association for Computational Linguistics, Brussels, Belgium (2018)

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 654–663. Association for Computational Linguistics, Melbourne, Australia (Jul 2018)