

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertervezés és -fejlesztés Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

Szerzőség-megállapítás magyar szóbeágyazási modellekkel mélytanulási alapokon

Szerző: Gulyás Oldal Laura
mérnökinformatikus MSc. szak, II. évf.
Konzulens: Dr. Kertész Gábor
egyetemi docens

Budapest, 2022

Tartalomjegyzék

Absztrakt	3
1. Motiváció és célkitűzés	4
2. Irodalomkutatás	6
2.1. Szerzőség-elemzés	6
2.1.1. Szerzőség-elemzés ágazatai	6
2.1.2. Szerzőség-megállapítás	7
2.2. Stílusjegyek	7
2.2.1. Lexikális stílusjegyek	7
2.2.2. Szintaktikai stílusjegyek	8
2.2.3. Szemantikai stílusjegyek	10
2.2.4. Tartalomfüggő stílusjegyek	10
2.2.5. Szóreprézenciációk	10
2.2.6. Magyar szövegelemző eszközök	14
2.3. Szövegelemző eljárások fejlődése	16
2.4. Megvizsgált mélytanuláson alapuló eljárások	17
2.4.1. Mélytanulós modellek architektúrái	17
2.4.2. Karakterszintű jellemzőkön alapuló eljárások	18
2.4.3. Szófajokon alapuló eljárások	22
2.4.4. Szóbeágyazásokon alapuló eljárások	24
2.4.5. Megvizsgált eljárások összehasonlítása	26
3. Eljárások kiértékelése magyar nyelvű szövegeken	29
3.1. Adathalmaz	29
3.2. Tanítás körülményei	32
3.3. Kiértékelt eljárások	33
3.3.1. Karakterszintű konvolúciós neurális hálózat	33
3.3.2. Szintaktikai rekurrens neurális hálózat	34
3.3.3. Szerzőség-megállapítás konvolúciós neurális hálózatokkal és szófajjegyzéssel	35
3.3.4. Kiértékelés eredményei	40
4. Összefoglalás	42
4.1. További kutatási tervek	42
Ábrajegyzék	44
Táblázatok jegyzéke	45

Absztrakt

A természetes nyelvfeldolgozás (*Natural Language Processing*, NLP) az informatikai tudományok egyik feltörekvő ágazata, feladata az elsődleges emberi kommunikációs eszköznek, a beszédnek, szabad szövegnek elemzése gépi módszerekkel. A természetes nyelvfeldolgozási megoldások egy része arra irányul, hogy egy szöveg vagy dokumentum szerzőjére vonatkozó tulajdonságokat állapítson meg, például szerző személye, kora, származása, stb. A szerzői stílust legjobban megragadó szövegelemek a karaktersorozatok, szófajok és különböző szóreprerzentációk.

A dolgozat elején részletesen bemutatásra kerülnek az angol nyelvű szövegek szerzőség megállapítására alkalmazható módszerek és tulajdonságaik. Ezt követően a jellemzők magyar szövegekből történő kinyerése kerül ismertetésre, illetve bemutatásra kerül milyen eszköztárak és metodikák alkalmazhatók erre a célra (szófajcímkézők, szövektorok és szóbeágyazási modellek). A magyar és angol nyelv morfológiai különbségei jelentik a legnagyobb kihívást az idegen nyelvre készült eljárások alkalmazására magyar nyelvű szövegeken, erre a problémára a dolgozatban új módszer kerül bemutatásra és kiértékelésre.

Magyar nyelvű szerzőség-elemzésre alkalmas adathalmazok hiánya miatt a feladat részeként annotált korpuszt is elő kellett állítani: az ebből a célból szépirodalmi szövegeket tartalmazó adathalmaz automatikus begyűjtésére, előfeldolgozására kidolgozott módszer részletes bemutatása szintén a dolgozat része.

1. fejezet

Motiváció és célkitűzés

A természetes nyelv az ember legfontosabb kommunikációs eszköze, így nem meglepő módon, a 21. századra jellemző egyre nagyobb számítási erőforrásoknak köszönhetően egyre széleskörűbbé vált a szövegelemző módszerek palettája. A megoldások egy része egy adott szöveg szerzőjének személyére vagy valamely tulajdonságára világít rá, ezt a tudományterületet nevezik szerzőség-elemzésnek. A szerzőség-vizsgálat már a számítástudomány rohamos fejlődését megelezően is nagy jelentőséggel bír, azonban az elemzésre manuális módszereket alkalmaztak az automatizált megoldások helyett. Ilyen jellegű törekvéssel már a 15. században is találkozhatunk, amikor is Lorenzo Valla humanista bebizonyította, hogy a *Donatio Constantini* adománylevél hamisítvány [1]. Ezen kívül, az egyik leghíresebb, a szerzőségvizsgálat témakörére kiterjedő vizsgálatok közé tartozik Shakespeare egyes műveinek szerzőségének tanulmányozása. Shakespeare számos drámájáról bizonyították be, hogy azok más szerzők közreműködésével íródtak, ilyen például a *The Noble Kinsman* és *VIII. Henrik* című drámák, melyeknek jelentős részét John Fletcher írhatta [2]. A két szerző együttműködése mindaddig csak nyelvészek feltételezése volt, amíg azt a szerzői stílusjegyeken végzett automatizált elemzések be nem bizonyították [3].

Az N leggyakoribb szó, mondathossz, szóhossz, szótagok hangsúly szerinti váltokozása, szókészlet gazdagsága a legegyszerűbb és a legkorábban használt stílusjegyek közé tartoznak. Ilyen stílusjegyek elemzésével vizsgálták és bizonyították a Shakespeare szerzőségére vonatkozó feltételezéseket [4]. A szerzőség-elemzést tehát először irodalmi szövegek szerzőinek vizsgálatára alkalmazták, azonban ezek a módszerek alkalmazhatók akár a bűnügyi nyelvészet, folyamatos hitelesítés (*continuous authentication*), plágiumellenőrzés és -detektálás, online üzenetek (e-mailek, blogok, fórumok) szerzőinek vizsgálatára is.

Stilometriának nevezik azt a tudományterületet, amely felöleli az összes stíluson alapuló szövegkategorizálási módszert, legyen az szerző, műfaj vagy akár korszak szerinti kategorizálás, így a stilometria és a szerzőség-elemzés fogalma gyakran használt együttesen. Hogy pontosan mit értünk „stílus” alatt és hogy a szöveg mely részei vagy tulajdonságai képezik a szerző stíluselemeit nem teljesen nyilvánvaló, ugyanis ez változhat az írott szöveg tematikájától, korszaktól, a szerző életkorától és egyéb más tényezőtől. A mesterséges intelligencia és a természetes nyelvfeldolgozás fejlődésével ezen a tudományterületen nagy áttörés tapasztalható, ugyanis ezek az eszközök teszik lehetővé az olvasás során észrevétlen jellegzetességek feltárását a jól ismert stíluselemek mellett. A gépi tanuláson alapuló hagyományos statisztikai módszerekkel rendkívül pontos eredmények érhetők el szerzőség-elemzés területén, viszont nagy

hátrányuk, hogy a modell felépítése erősen összefügg a bemeneti adatokkal. Ezzel szemben a mély neurális hálózatokon alapuló megoldások automatizáltan nyerik ki a szerzői ujjlenyomatot, így egy nagy tanítóminta esetén ilyen jellegű módszerekkel pontosabb eredmények érhetők el [4].

Angol nyelvű szövegekre számos megoldás született a szerzőség-megállapítás témakörében, azonban a magyar nyelven íródott szövegek stílusával foglalkozó tanulmányok száma rendkívül csekély. Az angol nyelvre készült megoldások nem alkalmazhatók közvetlenül magyar nyelvű szövegekre, ugyanis a magyar nyelv ábécéje eltér az angoltól, a szórend kialakítása egy mondaton belül, szavainak felépítése rugalmasabb, ezáltal az angolnál jóval összetettebbnek tekinthetők a magyar nyelvű szövegek [5].

A dolgozatomban gépi tanuláson alapuló módszereket vizsgálom meg a szerzőség-megállapítás témakörében, különösen nagy hangsúlyt fektetve a mély neurális hálózatokon alapuló rendszerekre. Az angol nyelvre készült nagy pontosságú modellek működését vizsgálom magyar nyelvű szövegekre, illetve oly módon valósítom meg, hogy azok magyar nyelvre is megfelelő eredményt nyújtsanak.

2. fejezet

Irodalomkutatás

2.1. Szerzőség-elemzés

Természetes nyelvnek azt a nyelvet nevezzük, mely egy közösségben, spontán módon szabadon alakult ki és fejlődött tovább [6]. Természetes nyelvfeldolgozás (*NLP*) az a tudományterület, amely annak vizsgálatával foglalkozik, hogy számítógépes erőforrások milyen módon használhatók fel a természetes nyelvű szövegek vagy beszéd megértésére és manipulációjára. Az *NLP* több diszciplína találkozásával alakult ki, melyek közé tartozik a számítástechnika, nyelvészet, matematika, pszichológia. Ezen kívül számos felhasználói célra alkalmazzák, mint például gépi fordítások (*machine translation*), természetes szövegek összegzése, keresztnyelvi információ-visszakeresés (*CLIR*), beszédfelismerés és más alkalmazási területek [7].

A szerzőség-elemzés az a folyamat, mely szöveges alkotások jellemzőinek vizsgálatával foglalkozik, azzal a céllal, hogy a szöveg szerzőjére vonatkozó következtetéseket vonhassunk le. Az alapjai a stilometriából, egy nyelvészeti tudományágból származnak, mely a szöveg stílusának statisztikai elemzését végzi [8].

2.1.1. Szerzőség-elemzés ágazatai

A szerzőség-elemzésnek (*Authorship analysis*) több ágazata alakult ki, ezek a következők [8, 9]:

- **Szerzőség-megállapítás** (*Authorship attribution* vagy *Authorship identification*): Szerzőség-megállapítással határozható meg annak a valószínűsége, hogy egy adott szöveget egy szerző írt egyéb általa írt tartalmak vizsgálatán keresztül.
- **Szerzőség-profilozás** (*Authorship profiling* vagy *Authorship characterization*): Szerzőség-profilozással a szerző profilja vagy valamely tulajdonságai határozhatók meg. A tulajdonságok lehetnek: kor, végzettség, kulturális háttér, stb.
- **Hasonlóság-detektálás** (*Similarity detection*): Hasonlóság-detektálás esetén két szöveg összehasonlításával határozzuk meg, hogy azok ugyanaz a személy által íródtak vagy sem anélkül, hogy a két szerzőt megpróbálnánk azonosítani. A hasonlóság-detektálást gyakran a szöveget több részre bontva végzik, ebben a formában a módszer alkalmazható plágiumdetektálásra vagy kollaboráció detektálására is.

A kutatásom során az ismertetett ágazatok közül a szerzőség-megállapítás témakörének megismerésére fektetek nagy hangsúlyt.

2.1.2. Szerzőség-megállapítás

A szerzőség-megállapítás szűkebb értelemben egy szerzőt rendel hozzá egy szöveghez, oly módon, hogy az értékészlet (összes szóba jöhető szerző) előre meghatározott és változatlan, valamint az összes szerzőnek valamennyi egyéb, a bemeneti szövegtől eltérő írása adott. A megoldandó probléma felfogható többsztályú klasszifikációnak. A feladatban a legnagyobb kihívást az jelenti, hogy olyan jellemzőket nyerjünk ki, melyek egy nagy szöveghalmazban változatlanok és azok eltérnek más szerzők szövegeiből kinyert leíróktól, ezeket a jellemzőket nevezik szerzői stílusjegyeknek [8].

2.2. Stílusjegyek

Stílusjegyeknek (*style markers*) azokat a szövegből kinyert stilisztikai jellemzőket nevezik, melyek egy szerző stílusát egyértelműen meghatározzák, ezeket együttesen szerzői ujjlenyomatnak nevezik. Szerző megállapításra a legmegfelelőbb jellemzők általában nem nyilvánvalóak és nem detektálhatók emberi szemmel, így maga a szerző sem tudja befolyásolni vagy tudatosan manipulálni azokat [10]. A szerzői stílusjegyeket négy fő kategóriába sorolhatjuk: lexikális, szintaktikai, szemantikai, tartalomfüggő stílusjegyek [8].

2.2.1. Lexikális stílusjegyek

A lexikális stílusjegyek tekinthetők a legegyszerűbb és legintuitívabb jellemzőknek. Használatuk mögött az az alapelgondolás, hogy a szövegre tekinthetünk úgy, mint tokenek¹ sorozatára, és ezeknek az egységeknek előfordulási gyakorisága jól meghatározza a szerzői stílust. Előnyük, hogy szabadon alkalmazhatók bármilyen nyelvre vagy korpuszra² tokenizer³ jelenlétében [12]. A lexikális stílusjegyek további kategóriákba sorolhatók [12, 8]:

- Karakterszintű:
 - Betűk aránya
 - Számjegyek aránya
 - Kis- és nagybetűs karakterek aránya
- Karakter n-gram⁴ szintű:
 - N leggyakoribb n-gram
- Szószintű:

¹Tokenizációnak nevezzük a szöveg felbontását kisebb egységekre, ún tokenekre. Tokenek lehetnek például karakterek, szavak, mondatok, stb.

²„meghatározott szempontok alapján kiválasztott szövegmennyiség, amelyen a nyelvész vizsgálatát végzi”[11]

³Tokenizernek nevezik azt az eszközt, amely a szöveg tokenizációját végzi

⁴n-gramoknak nevezünk n egymást követő tokent. Például a "szerző" szó a következő karakter 3-gramra bontható fel: "sze", "rző"

- Szavak átlagos hossza
- N leggyakoribb szó
- Szókészlet gazdagság
- Szó n-gram szintű:
 - N leggyakoribb szó n-gram
- Mondatszintű:
 - Átlagos mondat-hossz
- stb.

A fenti csoportosításokban és felsorolásokban néhány példát tüntettem fel, észrevehető, hogy a token-alapú lexikális stílusjegyek magasabb dimenziójú tokenek esetén is értelmezhetők (pl. bekezdések száma egy fejezetben belül, fejezetek száma a szövegrészletben belül, stb.).

Az N leggyakoribb szó az egyik legelterjedtebb lexikális stílusjegy, ugyanis bebizonyosodott, hogy a leggyakrabban használt szavak témától, a tartalom jellegétől függetlenül szerzőnként. Ezeket a szavakat nevezik „funkció” szavaknak („*function words*”). Ennek ellenére az N-leggyakoribb szó használata nem mindig célravezető, ugyanis az egyes szavakat ugyan abban az alakban használhatjuk több kontextusban is (például a „szerző” szónak eltérő jelentése van a következő mondatokban: „A szerző megírta a művet.”, „Az aranyérmes szerző személy első lett.”). Emellett, a bag-of-words⁵ szövegreprezentáció nem veszi figyelembe a szavak sorrendjét. Annak elkerülése érdekében, hogy ugyan az a szó más-más alakjában (képzők, ragok, jelek használata) is szerepeljen a reprezentációban, szükséges címszóképző⁶ eszköz (*lemmatizer*) a szótári alak kinyerése érdekében. Gyakran előforduló lexikális stílusjegyeknek tekinthetők még a szókészlet gazdagság (szókészlet és tokenek számának hányadosa) és a karakter n-gramok előfordulási gyakorisága [12].

2.2.2. Szintaktikai stílusjegyek

A szemantikai és szintaktikai stílusjegyek kinyerésére mélyebb, nyelvfüggő elemzések szükségesek. A szintaktikai stílusjegyek használatának alapötlete, hogy az emberek általában tudat alatt ugyan azzal a szintaktikai felépítéssel alkotnak mondatokat. Az ilyen jellegű stílusjegyek kinyerése morfológiai elemzésekkel végezhető el, az adott nyelvre kialakított szövegelemző (*parser*) segítségével. Ilyen eszközök nyelvtől függően eltérően működnek, így egyes nyelvek esetén szövegelemző hiányában nem nyerhetők ki a szintaktikai stílusjegyek [12]. Az alábbi felsorolásban szerepel néhány gyakran használt szintaktikai jellemző: szó-faj címkék, konstituens fa, függőségi fa.

⁵Bag-of-words szövegreprezentáció - a szöveget az egyes szavak és azoknak az előfordulási gyakoriságára képezzük le

⁶Címszó (*lemma*) - szótári szó, azaz, a különböző szóalakok egy együtvé tartozó csoportjának képviselője, „az azonos szótőből származó összes (általában azonos szó-fajú) szóalakot átfogó kategória” [13]. Például a „szomszédnak” szó címszója a „szomszéd”

Szófaj címkék

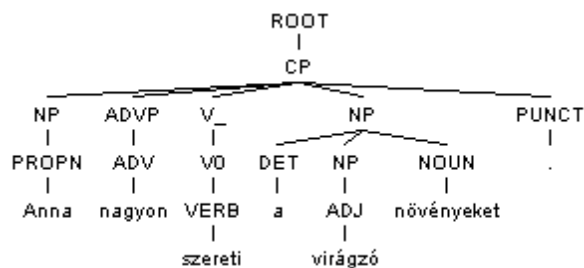
A szófajcímkék (*Part-of-Speech tags*, röviden: *POS tags*) a leggyakrabban használt szintaktikai jellemzők közé sorolhatók. Kinyerésükhöz szófajcímkézőre (*POS tagger*) van szükség. A szófajok értékei lehetnek például:

- főnév
- ige
- melléknév
- határozószó
- stb.

A stílusjegy használatának nagy hátránya, hogy a szófajok típusai, értékei nem egységesek különböző nyelveken. Ez annak tudható be, hogy különböző nyelveknél eltérőek a szófajok is, így például az angol nyelvű szövegeken kis számú címkével is jól jellemezhető a szó alakja. Morfológiailag gazdag nyelvek esetén, mint amilyen a magyar nyelv is, sokkal több adattal írható le a pontos alak, így ilyen alacsony számú címke nem használható fel szófajcímkézésre magyar nyelvű szövegeken. Ennek következménye, hogy az angol nyelvű bemenetekre tervezett módszer magyar nyelvű adatokon való alkalmazása esetén nem elegendő az angol szövegcímkéző leváltása magyar eszközre, az egész megtervezett modellt is át kell alakítani.

Konstituensfa

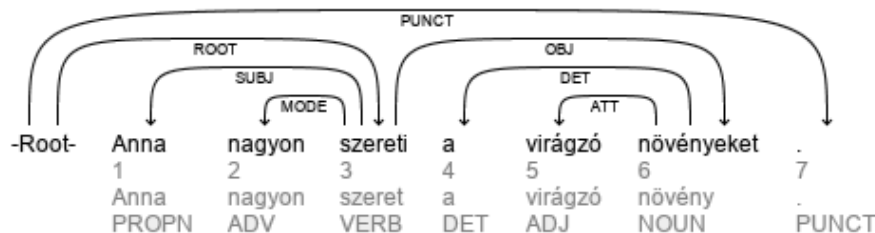
A konstituensfák (*Constituency tree*) a mondatok szavai között fellépő kapcsolatot írják le. Gyakran nevezik frázis struktúra fának is (*phrase structure tree*), frázisnak nevezzük azokat a szócsoporotokat, melyek együttesen töltenek be valamilyen szerepet a mondatban, például főnévi csoport (*Noun Phrase, NP*), igei csoport (*Verb Phrase, VP*). A konstituensfák főleg kötött szórendű nyelvekre eredményeznek jó mondatreprezentációt, így a magyar nyelvre inkább a függőségi fa használatos [14]. A 2.1 ábrán látható a „Anna nagyon szereti a virágzó növényeket.” mondat konstituensfája a magyarlanc konstituenselemző eszközzel kinyerve [15].



2.1. ábra. Egy magyar nyelvű mondat konstituensfája

Függőségi fa

A függőségi fa (*Dependency tree*) a mondat szavainak függőségi kapcsolatait írja le. Ebben a reprezentációban a struktúra gyökere a mondat fő igéje és minden más kapcsolati-egység (több szó esetén külön részfa) közvetlenül vagy közvetve kapcsolódik hozzá. A függőségi fa a konstituens fához képest egy pontosabb tartalmi elemzést végez [14]. A 2.2 ábrán látható a „Anna nagyon szereti a virágzó növényeket.” mondat függőségi fája a magyarlanc függőségelemző eszközzel kinyerve [15].



2.2. ábra. Egy magyar nyelvű mondat függőségi fája

2.2.3. Szemantikai stílusjegyek

A szemantikai elemzés az a folyamat, mellyel értelmet, jelentést társítunk a szavak sorozatához, mondatstruktúrához. Szemantikai elemzések során nem csak a mondat szerkezetét, hanem az egyes szavak szemantikai kapcsolatait is kinyerjük. Az egyik leggyakrabban használt, szemantikai elemzéssel kinyert mondatreprezentáció a szemantikai függőségi gráf [12].

2.2.4. Tartalomfüggő stílusjegyek

A tartalomfüggő stílusjegyek a bemeneti szövegek típusától függenek. E-mailek szerzőségi-elemzése során jó stílusjegyeknek tekinthetők a szöveg struktúrális jellemzői, például üdvözlési formák, bekezdések hossza, behúzások, stb. Amennyiben az összes rendelkezésre álló adat ugyan arról a témakörrel szól, jó stílusjegyek tekinthetők a szövegre vonatkozó kulcsszavak előfordulása is [12].

2.2.5. Szóreprezentációk

Az előző fejezetek a szerzőség-elemzés során alkalmazott stílusjegyeket részletezik. A szavak szöveges formájukban nem értelmezhetők és alkalmazhatók a modernebb, gépi tanuláson alapuló megoldások által, azokat vektorokként szokás reprezentálni. Ebben az alfejezetben ismertetem ennek módszereit.

One-hot kódolás

A szavak vektorosítására megoldást nyújthat a szavak one-hot vektor reprezentációja. A one-hot kódolás a gépi tanuláson alapuló módszerek során gyakran használt eljárása, mely egyszerűen az adott bemeneteket kategórikus adatokká képezi le oly

módon, hogy a korpuszban előforduló n egyedi szó esetén n -dimenziós vektort rendel hozzá minden szóhoz. Tehát, abban az esetben ha a korpuszunk 3 egyedi szóból áll (például *alma*, *ágy*, *körte*), akkor a szavakat 3-dimenziós vektorokra képezi le, ahol az egyik koordinátájának értéke 1 lesz, míg az összes többinek 0 (például az *alma* szó one-hot vektora $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, míg az *ágy*, *körte* szó one-hot vektorai $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ és $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$). A one-hot kódolás legnagyobb hátránya a dimenzionalitás, ugyanis egy 50 000 egyedi szóból álló korpusz esetén 50 000-dimenziós vektorokként reprezentálja az egyes szavakat, ami nagyon nagy tárhelyigényt jelent kevés információt hordozó adatra. Másik nagy hátrányuk, hogy a leképezett vektorok távolságai nem utalnak a szavak szemantikai jelentésére [16]. Például az *alma* és *körte* szavak szemantikai értelemben hasonlóak és azonos kontextusban használjuk azokat. Ennek ellenőre a távolságuk azonos a *alma* és *ágy* szavak vektorainak távolságával, melyek szemantikai értelemben jobban eltérnek az előbb említett párostól. A következő alfejezetekben mutatok be másik két megközelítést, melyek kiküszöbölik ezeket a problémákat.

Disztribúciós szemantikai modellek

Használatuk mögötti alapfelgondolás, hogy az egyes szavak szemantikai jelentése nagyban függ attól, hogy milyen kontextusban használjuk azokat, azaz milyen szavak találhatók a szó egy előre meghatározott környezetében. Például a *Ma reggel ettem egy almát.* és a *Ma reggel megkóstoltam egy almát.* mondatokból arra lehet következtetni, hogy az *ettem* és *megkóstoltam* szavak szemantikai értelemben hasonlóak, ugyanis azonos kontextusban használjuk őket. Mindkettő szót a *ma*, *reggel*, *egy*, *alma* szavak környezetében használjuk. A modell a valóságban a *(ma, alma)*, *(reggel, alma)*, *(egy, alma)*, stb. szópárok előfordulását vizsgálja a korpuszon. A disztribúciós szóreprezentációval leképezett vektorok közötti távolság kicsi lesz, így a one-hot kódolásnál felmerült egyik hiányosság ezzel a módszerrel kiküszöbölhető. A módszer nagy hiányossága, hogy nem tünteti el a nagy dimenzionalitás problémáját, hiszen általában a szavak a korpusz méretéhez képest kevés szóval szerepelnek együttesen és emiatt „ritka” vektorokat eredményez a reprezentáció, azaz nagydimenziós vektorokat, melynek legtöbb koordinátája 0 [17].

Szóbeágyazási modellek

A szóbeágyazási modellek (*word embedding*) neurális hálózatokon alapuló eljárások, melyek kiküszöbölik a nagy dimenzionalitás problémáját, használatukkal tömör vektorreprezentációt kapunk eredményül. A kapott vektorok alacsony dimenziójúak és koordinátáik általában 0-tól eltérő valós számok $[-1, 1]$ intervallumon. A hasonló jelentésű szavakat általában hasonló környezetben használjuk, így a hasonló jelentésű szavak közötti távolság kicsi lesz [17].



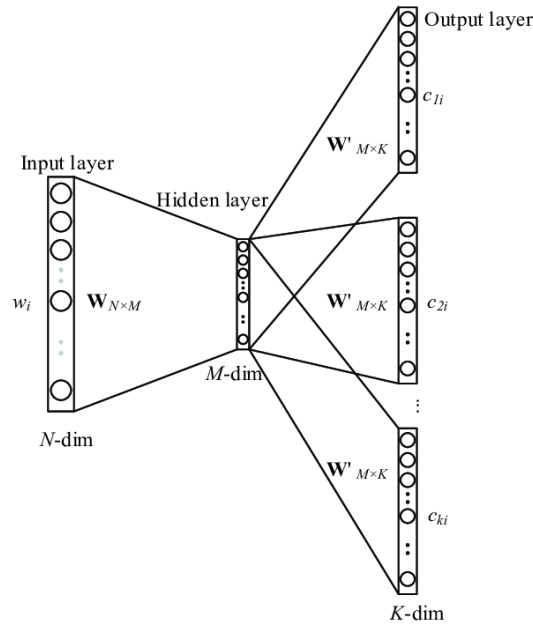
2.3. ábra. A *reggeli* szó környezete a vektortérben 2-dimenzióra projektálva [17]

A 2.3 ábrán látható a *reggeli* szó környezete a vektortérben. Ez a szó kétértelműsége miatt érdekes, az ábra jobb felső sarkában olyan szavak találhatók, melyek a reggelihez, mint étkezéshez hasonló szöveggörnyezetben fordulnak elő a korpuszban, míg a bal alsó sarokban a reggeli, mint napszakhoz hasonló jelentésű szavak jelennek meg. Fontos megjegyezni, hogy a szóbeágyazások szóvektorai nem a szó jelentésére vonatkozó információt hordozzák, hanem a hasonló módon használt szavakat rendezi csoportokba. Emiatt, ahogyan a 2.3 ábrán is megfigyelhető, a *reggel* és *este* szavak ellentétes jelentésűek, viszont a vektortérben közel helyezkednek el egymáshoz.

Word2Vec

A Word2Vec egy olyan eljárás, mely lehetővé teszi a szóbeágyazási modellek tanítását. A módszert Mikolov és társai [18] tervezték és valósították meg. Olyannyira elterjedtté vált, hogy a szóbeágyazási modelleket és a Word2Vec fogalmakat az irodalomban együttesen használják, azonosnak veszik. Az eljárás a célszó vektoros reprezentációját állítja elő a környező szavakat felhasználva mesterséges neurális hálózatokkal [19].

2-féle eljárással végezhető el a modellek tanítása: CBOW (*Continuous Bag of Words*) és Skip-gram modell. A két modell közötti legnagyobb különbség a bemeneti és kimeneti adatok jellege. A Skip-gram modell esetén a hálózat bemenete a célszó, míg a kimenete a célszó egy előre meghatározott fix környezetében található szavak halmaza. Ez a CBOW modell esetén fordítva történik: a hálózat bemenetei a környező szavak és a kimenete pedig a célszó. A Skip-gram modell lassabban tanul, azonban a ritkán előforduló szavakat pontosabban helyezi el a vektortérben a CBOW modellhez képest.



2.4. ábra. Skip-gram modell felépítése [20]

A Skip-gram modell felépítése látható a 2.4 ábrán. Mindkét modell esetén az eljárás először az összes előforduló szó alapján felépít egy szótárat, melynek elemszáma N . A Skip-gram modell a célszó N -dimenziós one-hot vektoros reprezentációját kapja bemenetül, míg a modellnek átadott kimenetei a környező szavak N -dimenziós one-hot vektoros reprezentációi. A hálózatnak rejtett rétege M -dimenziós, ahol az M a szóbeágyazások dimenziója.

Először a modell a rejtett réteg súlyait véletlenszerűen inicializálja, kiszámolja a keresztentropia hibát, majd visszaterjeszti azt oly módon, hogy a modell kimenete a kiszámolt súlyokkal a környező szavakat prediktálja. A modell betanítását követően a rejtett réteg súlyai lesznek a szavak M -dimenziós szóbeágyazásai.[18]

fastText

Morfológiailag gazdag nyelvek esetén, mint amilyen a magyar, ugyan az a szó számos alakjában fordulhat elő a ragok, képzők, jelek használatával. Emellett, egyes nyelvekre sokkal inkább jellemző az összetett szavak használata. Emiatt, a Word2Vec bizonyos nyelvek esetén nem teljesít megfelelően, szórészlet alapján nem tudja elhelyezni a vektortérben azokat a szavakat, melyek nem szerepeltek az eredeti szótárban. Erre a célra találtak ki a fastText eljárást, mely a Word2Vec egy kiterjesztése.[21, 22]. Az eljárás lényege, hogy a bemeneti szavakat karakter n -gramokra bontja és azokon végzi a tanítást [21]. Például, ha 3-gramokon végzi a tanítást az eljárás és Skip-gram modellel tanítjuk a szóbeágyazásokat, akkor a bemeneti adatok a „téved” szó esetén a „tév”, „éve”, „ved” szórészletek lesznek. Ennek köszönhetően, az eljárás a „téved” és „összetéveszt” szavakhoz közeli vektorokat fog hozzárendelni akkor is, ha a „téved” szó már szerepelt a szótárban és a „összetéveszt” szó még nem.

Szóbeágyazások használata transfer learning-gel

A mély neurális hálózatok használatának nagy korlátja, hogy nagy mennyiségű tanítási adatra van szükségünk ahhoz, hogy a hálózat megtalálja a problémára

specifikus összefüggéseket. Erre találták ki a transfer learning-ket melynek alap-gondolata, hogy egy sokkal általánosabb célra betanított hálózat súlyait fel tudjuk használni a specifikusabb célra kialakított hálózat kezdeti súlyainak beállítására. [23]

Például, a „néz”, „szemlél”, „bámul” szavak bármilyen szöveggörnyezet esetén hasonló jelentéssel bírnak, attól függetlenül hogy a szóbeágyazási modellt hírportálok cikkein, gyerekmeséken, szakmai jellegű cikkeken vagy szépirodalmi szövegeken tanítjuk. Ilyen modelleket felhasználva, el tudjuk érni, hogy az aktuális modellünket ne kelljen teljesen az elejéről betanítanunk ha nem áll rendelkezésünkre elég adat. Ez után tovább taníthatjuk a szóbeágyazási modellt, hogy az a szépirodalmi szövegekre adaptálja a szóbeágyazások súlyait [24]. Ilyen előretanított szóbeágyazási modellek léteznek magyar nyelvre is, mint például a *fastText wikipedia* vagy *Szeged word vectors* [25]. A dolgozatomban az utóbbit használtam, ugyanis a wikipedia jelentős mennyiségű politikai, történelmi kontextusú cikket tartalmaz, ahol a szóhasználat nagymértékben eltérhet a szépirodalmi szövegekben használt nyelvezettől. A *Szeged word vectors* szóbeágyazási modellt 4.3 milliárd tokenen tanították [25], a Word2Vec és fastText modellek közül az utóbbit használtam a fastText modell előző fejezetben ismertetett előnyei miatt.

2.2.6. Magyar szövegelemző eszközök

A kutatásom részeként megvizsgáltam milyen szintaktikai elemzést végző magyar nyelvre kifejlesztett eszköztárakat vizsgáltam meg. Ennek legfőbb oka, hogy a szerzőség-elemzés témakörében megvizsgált eljárások, melyeket magyar nyelvre is megvizsgáltam szintaktikai vagy karakterszintű stílusjegyeket használnak fel.

A legtöbb magyar nyelvre kifejlesztett szövegelemző eszköztár legnagyobb hátránya, hogy eltérő morfológiai kódolórendszert használ. Ez abban nyilvánul meg, hogy a morfológiai elemzés során más eszköztárak más címkéket rendelnek hozzá az egyes szavakhoz, különbözik a címkézésük. Ilyenek például a Humor, MSD és KR kódolórendszerek [15]. A *hunmorph* morfológiai elemző a KR⁷ kódolórendszert használja, míg a *hunpos* szófajcímkéző MSD⁸ kódolórendszert alkalmaz. A magyar-lanc szintaktikai szövegelemző eszköztár a *Universal Dependencies and Morphology* röviden: UD formátumra alakítja át az annotációkat. Az UD projekt egységes címkérendszert alakított ki számos nyelvre annak érdekében, hogy nyelvfüggetlen szintaktikai mondatrepresentációt nyerhessünk ki a szövegből [26].

Szintaktikai szövegelemzést végző eszköztárakat treebankek felhasználásával valóstítják meg. Treebankeknek nevezik a szintaktikai és szemantikai címkékkel ellátott korpuszokat. Az egyetlen manuálisan címkézett magyar nyelvű treebank a Szeged Treebank. A Treebank MSD címkézési adattal annotált adatokat tartalmaz, azonban olyan konvertereket fejlesztettek ki, mellyel az MSD címkék leképezhetők UD címkékké, így elérték azt, hogy magyar nyelvű szintaktikai elemzők is nyelvfüggetlen címkékkel jelölhessék a mondatrészeket [26].

⁷magyar nyelvre kialakított kódolórendszer

⁸kelet-európai nyelvekre kialakított egységes kódolórendszer [26]

Magyarlanc

A *magyarlanc* szövegelemző eszköztár szintaktikai elemzést végző eszközök gyűjteménye és azoknak egy továbbfejlesztett egységesített változata magyar nyelvre. Az eszköztár a következő feladatok ellátására képes [15]:

- Szöveg mondatokra bontása
- Tokenizáció
 - Szófajcímkézés
 - Lemmatizálás
- Konstituens elemzés
- Függőségi elemzés

A magyarlanc szófajcímkéző modulja a *purePOS* eszköz egy módosított változata, amely a KR kódolórendszer címkéivel végzi az annotálást, ezeket utólag alakítja át UD formátumra. A függőségi elemzést végző modul a *Bohnet parser*, míg a konstituens elemző modul a *Berkeley parser* magyar nyelvre adaptált változata [15].

A magyarlanc eszköztár 3.0 verzióját használtam a kutatásomban. A szoftver egy JAVA nyelven íródott alkalmazás, melynek elindítása parancssorból lehetséges a bemeneti és kimeneti állományok elérési útját és a futtatás módját megadva. A módok, melyek támogat az eszköztár a *constparse*, *depparse*, *morphparse*, *parse* és *gui*. *Gui* módban az alkalmazás rendelkezik felhasználói felülettel és az elemezni kívánt szöveget közvetlenül lehet bemásolni az alkalmazás egyik szövegdobozába. Ebben a módban az elemzésnek a kimenete nem csak egy szöveges állomány, hanem a kielemezett szöveg függőségi és konstituens fája is (például a 2.1, 2.2 ábrák). A többi módban a program kimenetei szöveges állományok, ahol a kiírt elemzési szempontok a kiválasztott módtól függően változnak. A kutatásomban a *depparse* módot használtam. A program kimenete a „- Rajta! Vili már futott.” szövegű bemenetre a 2.1 táblázatban látható.

1.	2.	3.	4.	5.	6.	7.
1	-	-	PUNCT	—	0	PUNCT
2	Rajta	rajta	INTJ	—	0	ROOT
3	!	!	PUNCT	—	0	PUNCT
1	Vili	vili	NOUN	Case=Nom Number=Sing	3	SUBJ
2	már	már	ADV	—	3	TLOCY
3	futott	fut	VERB	Definite=Ind Mood=Ind Number=Sing Person=3 Tense=Past VerbForm=Fin Voice=Act	0	ROOT
4	.	.	PUNCT	—	0	PUNCT

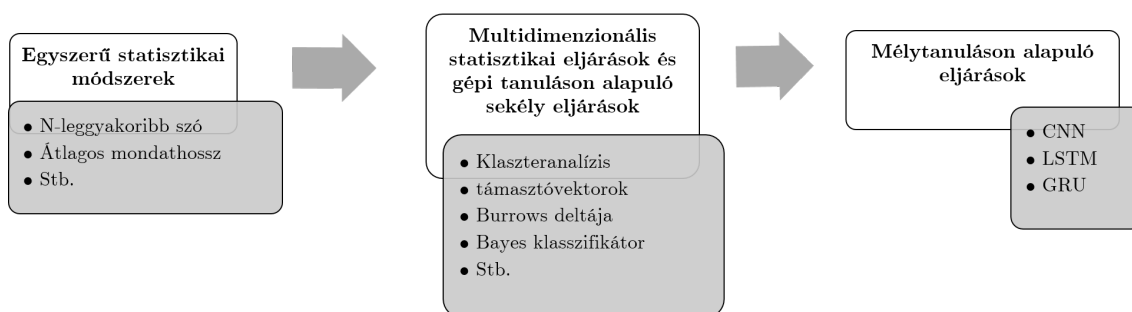
2.1. táblázat. magyarlanc *depparse* módú elemzésének kimenete

A függőségi elemzés kimeneti állományában az egyes mondatok között üres sorok szerepelnek. Az állomány minden nemüres sora egy szó szintaktikai jellemzőit tartalmazza tabulátorral elválasztva. Az egyes oszlopokban a következő jellemzők szerepelnek [15]:

1. Szó sorszáma a mondatban
2. A szó eredeti alakja

3. A szó szótári alakja (*lemma*)
4. Szófaj (*POS tag*)
5. Morfológiai jellemzők
6. Szülőcsomópont azonosítója
7. Függőségi címke⁹

2.3. Szövegelemző eljárások fejlődése



2.5. ábra. Szövegelemző eljárások fejlődése

A természetes nyelvfeldolgozásban használt eljárások a gépi tanulás megjelenésével rohamos fejlődésen mentek keresztül. A kezdeti megoldások jelentős része statisztikai módszereken alapulnak, melyek a szövegből kinyert lexikális jellemzőket használják fel szerzőség-elemzésre. Eleinte olyan megoldások születtek, melyek egyetlen lexikális jellemzőt vizsgálva végzik az elemzést (például átlagos mondathossz, szókészlet gazdagság, stb.), ez látható a 2.5 ábra baloldali dobozában. Később jelentek meg multidimenzionális statisztikai módszerek, melyek komplexebb és ezáltal pontosabb elemzést végeztek. Ilyen eljárások közé tartozik a *klaszteranalízis*, *Support Vector Machines*, *Nearest Shrunk Centroids*, *Burrows Deltája*, mint ahogyan az a 2.5 ábra középső dobozában látható. A multidimenzionális eljárások esetén a stílusjegyeket vektor formájában reprezentálják és az azok közti távolság határozza meg a két szöveg stilisztikai hasonlóságát. Egy ilyen megközelítés esetén szükséges a jellemző szerkezetének pontos ismerete, illetve olyan távolságmérték (*euklideszi*, *Manhattan*, *Classic Delta*, *Burrows Delta*, *Canberra*, stb.) definiálása, mely a szövegek közötti különbséget jól leírja. Szükséges a korpusz pontos ismerete, a megfelelő jellemzők kiválasztása és az ehhez illeszkedő távolságmérték meghatározása [4]. Emiatt, a kialakított modell nagyon érzékeny és rugalmatlan az összes előbb említett tényező miatt.

A mély neurális hálózatok maguk fedezik fel a pontos összefüggéseket a jellemzők között, így használatukkal rugalmasabb, robusztusabb eljárások érhetők el [4]. Az *SVM* és *naív Bayes klasszifikátor* (NBC) a szövegklasszifikációban a leggyakrabban használt sekély eljárások közé sorolhatók [28]. Az eljárások nagy sikert élveztek a

⁹a szó mondatban betöltött szerepe vagy kapcsolata a szülőcsomóponttal [27]

szerzőség-elemzés területén, azonban a jellemzők pontos megtervezése nagy akadályokat okozott a további előrehaladások elérése érdekében. Ezeket a nehézségeket küszöbölik ki a mély neurális hálózatok, ahol nincs szükség a jellemzők megtervezésére, a hálózat maga tanulja meg az adatokból a megfelelő reprezentációt a sok rejtett réteg segítségével [29]. Amellett, hogy a mély neurális hálózatokkal a jellemzők pontos megtervezése nem szükséges, elmondható, hogy jelentősen nagyobb pontosság érhető el szövegklasszifikációs feladatokra ilyen megközelítéssel, mint *SVM*-ekkel vagy *naïv Bayes klasszifikátorral* [28]. A kutatásom során mély neurális hálózatokkal megvalósított eljárásokat vizsgállok meg a szerzőség-megállapítás problémájára.

2.4. Megvizsgált mélytanuláson alapuló eljárások

Ebben a fejezetben fogom részletezni, hogy milyen korszerű, mélytanuláson alapuló megoldások születtek a szerzőség-megállapítás problémájára. Az általam megvizsgált módszereket a felhasznált stílusjegyek és a neurális hálózat architektúrája szerint fogom csoportosítani, ezek a következők:

1. **Karakterszintű jellemzőkön alapuló eljárások**

Architektúra: Konvolúciós neurális hálózatok [30, 31, 32]

2. **Szófajokon alapuló eljárások**

Architektúrák: LSTM és Konvolúciós neurális hálózatok [33, 34]

3. **Szóbeágyazásokon alapuló eljárások**

Architektúrák: LSTM és GRU hálózatok [35]

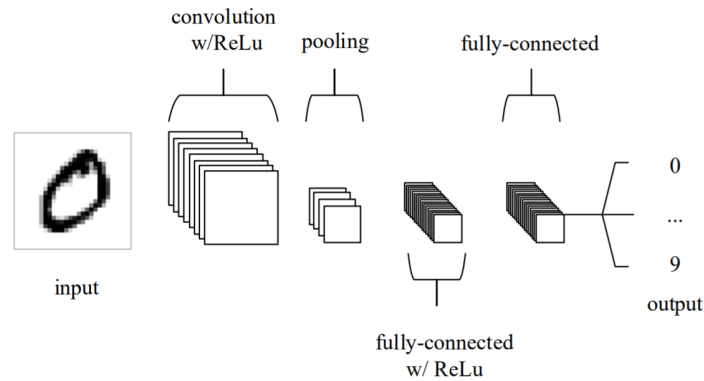
Mint ahogy az előző csoportosításban látható, a megvizsgált eljárások a konvolúciós, LSTM és GRU neurális hálózatok architektúráját alkalmazzák. A következő fejezetben ezeknek a működését fogom röviden leírni, míg a megvizsgált eljárások működési elvét az azt követő fejezetekben ismertetem.

2.4.1. Mélytanulásos modellek architektúrái

A neurális hálózatok forradalmában számos architektúra alakult ki, melyek közül a legelterjedtebbek közé sorolhatók a konvolúciós és rekurrens neurális hálózatok. Nagyszámú, a természetes nyelvfeldolgozás területén előforduló mélytanulási eljárás ezeken az architektúrákon alapszik, így ezeknek a működését ismertetem a következő alfejezetekben.

Konvolúciós neurális hálózatok

A konvolúciós neurális hálózatokat (*Convolutional neural networks*, röviden: CNN) elsődlegesen mintafelismerés (*pattern recognition*) területén használják. Általában képi adatokra szokás alkalmazni ezt az architektúrát, azonban egyre többet használnak egydimenziós adatokra is, mint amilyen a hang vagy folyó szöveg [36].



2.6. ábra. Egyszerű konvolúciós architektúra [36]

A konvolúciós hálózatok általános felépítése látható a 2.6 ábrán. A hálózat első rétege a konvolúciós réteg, ahol a modell konvolúciós műveleteket végez a rétegben használt kerneleken. A réteg kimenetei az ún. aktivációs térképek, ezek lesznek az összevonó (*pooling*) réteg bemenetei, melynek feladata az aktivációs térkép paraméterszámának csökkentése. A hálózat további rétegei teljesen kötött rétegek ReLu aktivációs függvényvel [36]. A konvolúció egyszerű algebrai műveletekből tevődik össze, mint a szorzás és összeadás, emiatt a konvolúciós modellek tanítása rendkívül jól párhuzamosítható.

Rekurrens, LSTM és GRU hálózatok

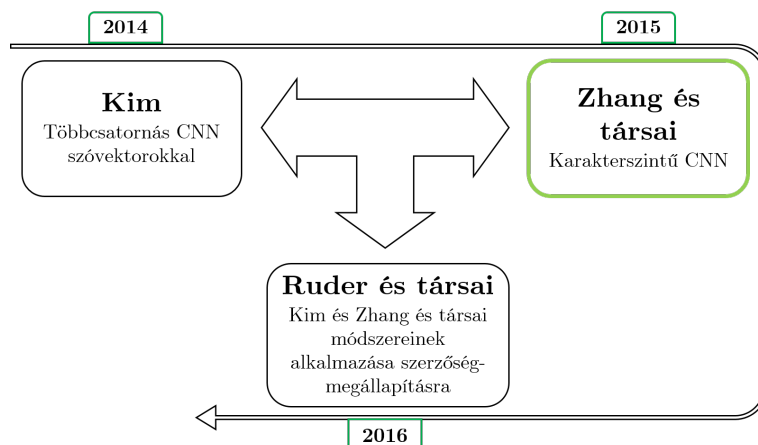
Olyan területeken, ahol a bemeneti adatok időbeli változása az adat egyik legfontosabb tulajdonsága, a hagyományos előrecsatolt neurális hálózatok nem teljesítettek kielégítően. Ezt a problémát oldották meg a rekurrens neurális hálózatok (röviden: RNN), ahol a bemeneti adat szekvencia vagy idősor. Idősoros adatoknak tekinthetők a hang, szöveg, kézírás, így az RNN architektúrát a beszédfelismerés, természetes nyelvfeldolgozás területén alkalmazzák. A rekurrens hálózatok nagy hátránya, hogy csak rövidtávú függőségeket tud megjegyezni az eltűnő és robbanó gradiens jelenség miatt [37]. Erre a problémára találták ki az LSTM (*Long Short Term Memory*) hálózatokat, ahol az LSTM cellák a hosszútávú függőségeket is megjegyzik. Az LSTM cellákban az ún. kapuk felelnek azért, hogy eldöntsék, mely információt érdemes tovább engedni a következő rétegekbe, tehát milyen adatot érdemes hosszú távon megjegyezni.

A GRU (*Gated Recurrent Unit*) hálózatokat az LSTM-hez hasonló adatokra és alkalmazási területekre tervezték, azzal a különbséggel, hogy a GRU egységek felépítésükben jóval egyszerűbbek az LSTM-nél, azonban teljesítményük hasonló [37]. A CNN hálózatokhoz képest az LSTM és GRU hálózatok tanítása jelentősen lassabb, ugyanis a rekurrens architektúrák lényege, hogy az idősoros adatot a szekvencia sorrendjében, szekvenciálisan dolgozza fel. Emiatt, a rekurrens hálózatok műveletei kismértékben párhuzamosíthatók, így tanításuk is jóval lassabb a CNN hálózatokhoz képest, ahol az egyes műveletek és adatok teljesen függetlenek egymástól.

2.4.2. Karakterszintű jellemzőkön alapuló eljárások

A konvolúciós hálózatok gyakran használt eljárások a szerzőség-elemzés problémájára. A hálózat konvolúciós rétegeinek szűrői olyan bemenetekre aktiválódnak, me-

lyeknek valamely része a tanítási adathalmazra jellemző mintát tartalmaz. A konvolúciós hálózatok képi adatok feldolgozásában arattak nagy sikert, azonban egydimenziós adatokra is kiválóan alkalmazhatók, minták efféle adatokon is felfedezhetők [31]. Karakterszintű konvolúciós hálózatok esetén a szöveget nem idősoros adatként kezeljük, hanem karakterek sorozataként. Szövegklasszifikációra először Zhang és társai [31] alkalmazták a karakterszintű konvolúciós neurális hálózatokat.



2.7. ábra. Karakterszintű konvolúciós hálózatok fejlődése

Zhang és társai, 2014: Karakterszintű konvolúciós hálózatok mondatklasszifikációra

Az eljárás publikálásakor léteztek konvolúciós hálózaton alapuló mondatklasszifikációs megoldások, melyek bizonyítottan jó eredményeket produkáltak. Egyik olyan módszer, amely nagy áttörést okozott a természetes nyelvfeldolgozásban Yoon Kim módszere, aki többsatornás konvolúciós modellt javasolt mondatklasszifikációra, mint ahogyan az a 2.7 ábrán látható. Kim kétsatornás architektúrát alkalmazott, az egyik csatorna bemenetei statikus míg a másik csatorna bemenetei nem statikus szóbeágyazások¹⁰ [30]. A másik nagy különbség Zhang [31] és Kim módszere között a bemeneti adatok jellege, Kim módszerének bemeneti adatai szóbeágyazások, míg Zhang és társainak eljárásának bemenetei karakterek. Zhang és társai [31] voltak az elsők, akik konvolúciós hálózatokat alkalmaztak karaktereken mondatklasszifikációs problémára.

Az eljárás felépítéséből és az adatok jellegéből elmondható, hogy a modell mintákat a karakterek sorozatában keres. A konvolúciós hálózat a többféle jellemzőt is meg tud ragadni: például milyen gyakoriak a szóközök, azaz milyen hosszúak a mondatok, vannak-e ismétlődő szomszédos karakterek az egyes szavakban (például a „szállítóvállalat” szóban az 'l' betű kétszer is duplán fordul elő), a használt szavakban milyen sűrűn fordulnak elő gyakori megánhangzók, stb.

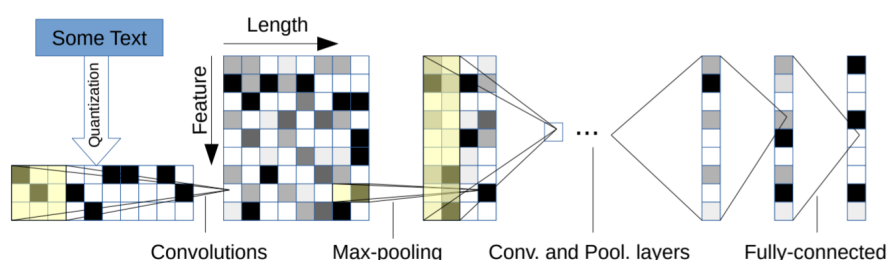
Két új modellt javasoltak, egyet nagyobb és egyet kisebb méretű konvolúciós filterrel, ezeket megvizsgálva két további szempont szerint: eltérő adatként kezeljük-e a nagybetűs és kisbetűs karaktereket vagy sem. Az eljárásokat összehasonlította más szövegklasszifikációs módszerekkel ugyan azokon az adathalmazokon, beleértve a meglévő, nem karakterszintű konvolúciós modelleket és a természetes nyelvfeldolgozásban használt legelterjedtebb statisztikai módszereket, mint a bag-of-words és

¹⁰szóbeágyazások az adathalmazra finomhangolt változatai

bag-of-ngrams modell.

Zhang és társai által javasolt modell felépítése látható a 2.9 ábrán, míg a modell rétegeire vonatkozó paraméterek az 2.2 táblázatban láthatók. Zhang és társai egy kis- és egy nagyméretű modellt javasolnak, az előbbi esetén a konvolúciós réteg mérete 256, míg az utóbbi esetén 1024.

A módszerben használt szövegrészlet hossza 1014 karakter. Ennek következtében a bemeneti adatok egy olyan 2-dimenziós struktúrára vannak leképezve, ahol az egyes sorok a karakterek one-hot vektorral kódolt formái (a használt karakterkészlet esetén 70-dimenziós vektor). Több eltérő méretű konvolúciós szűrőt alkalmaztak, ahogy az a 2.2 táblázat 4. oszlopában látható. Az 5. oszlopban az összevonó (*max-pooling*) rétegek mérete látható. A modell első 6 rétege konvolúciós rétegek, míg az utolsó kettő teljesen kapcsolt rétegek. Ebben a két rétegben 0.5 értékű dropout mechanizmust alkalmaznak a túlillesztés elkerülése érdekében.



2.8. ábra. Zhang és társai által javasolt konvolúciós modell felépítése [31]

Réteg	Nagy	Kis	Kernel	Összevonó réteg mérete
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3
Kimeneti egységek		Kimeneti egységek		
7	2048	1024		
8	2048	1024		
9	Osztályok száma			

2.2. táblázat. Zhang és társainak modelljének rétegeinek paraméterei

Az eljárásokat 6 különböző adathalmazon értékelték ki. Az adathalmazok két nagy csoportba oszthatók:

- újságcikkeket vagy fórumokon feltett kérdéseket tartalmazó adathalmaz, ahol az osztályok a szövegre vonatkozó szövegekategoróriák, témakörök (osztályok száma: 4-14)
- szöveges értékeléseket tartalmazó adathalmazok, melyeknek osztályai az értékelések számok formájában, például 1-től 5-ig (osztályok száma: 4-5).

A bemeneti adatok mindegyikének hossza 1014 karakter, a legkisebb adathalmaz teljes mérete 127 600, ebben az esetben a tanítás iterációinak száma (angolul: *epoch*)

5 000 volt. A legnagyobb adathalmaz mérete 3 650 000, a tanítás 30 000 iteráción keresztül történt.

A legpontosabb eljárás megválasztásánál az eljárások hibaarányait vették figyelembe. A kisebb méretű¹¹ adathalmazokon a legjobban a hagyományos statisztikai módszerek teljesítettek, míg a nagyobb méretű¹² adathalmazokon a karakterszintű konvolúciós hálózatok végezték a legpontosabb osztályozást. A modellek különböző változatai más-más adathalmazokon eredményeztek kisebb hibaarányt, így adathalmazfüggő melyik variációt célszerű használni. Érdeemes megjegyezni, hogy szóbeágyazásokon alapuló konvolúciós hálózatok a kisebb méretű adathalmazokon jobb eredményt nyújtottak mint Zhang és társainak módszere, azonban a statisztikai hagyományos módszerek bizonyultak a legpontosabbnak ezeken az adatokon. Nagyméretű adathalmazokon Zhang és társainak módszere biztosította a legkisebb hibaarányt az összes megvizsgált eljárás közül [31].

Ruder és társai, 2015: Karakterszintű és többcsatornás konvolúciós neurális hálózatok nagyszabású szerzőség-megállapításra

Ruder és társai voltak az elsők, akik a Zhang [31] és Kim [30] által javasolt szövegklasszifikációs megoldásokat szerzőség-megállapításra alkalmazták, ahogyan az a 2.7 ábrán látható. Publikációjukban a két eljárás különféle változatait hasonlítják össze. A megvizsgált eljárások a következők [32]:

- **CNN-szó:** Egycsatornás CNN nem statikus szóbeágyazású bemenettel
- **CNN-karakter:** Egycsatornás karakterszintű CNN
- **CNN-szó-szó:** Kétszatornás CNN egy nem statikus és egy statikus szóbeágyazású csatornával
- **CNN-szó-karakter:** Kétszatornás CNN egy nem statikus szóbeágyazású csatornával és egy karakterszintű csatornával
- **CNN-szó-szó-karakter:** Háromcsatornás CNN egy nem statikus és egy statikus szóbeágyazású csatornával, valamint egy karakterszintű csatornával

Az eljárások kiértékelése során a következő típusú adatokon végezték a szerzőselemezést: emailek, filmes értékelések, blogok, Reddit bejegyzések és hírességek Twitter bejegyzései.

Az adathalmaz mérete rendkívül nagy, nagyszámú osztállyal. A legkisebb adathalmaz mérete 80 852, 144 osztállyal, míg a legtöbb adattal rendelkező adathalmaz mérete 2 000 000, 9 266 osztállyal, a tanítást és a modellek kiértékelését azonban az adathalmazok töredékén végezték. Az eljárások átfogó összemérése 10 és 50 szerzős adathalmazon történt. A későbbiekben további, sokszerzős adathalmazon is megvizsgálták a legjobbnak bizonyult eljárásokat. Az adatok maximális hosszának az 500 szót vagy 3000 karaktert határozták meg.

Az eljárások összehasonlításakor a legnagyobb F1¹³ értéket eredményező módszereket tekintették a legjobbnak. Az összes megvizsgált konvolúciós modell közül

¹¹kisebb méretű adathalmazok mérete: 127 600 - 600 000

¹²nagyobb méretű adathalmazok mérete: 650 000 - 3 000 000

¹³F1 érték - precizitás és szenzitivitás harmonikus közepe

a CNN-karakter eljárás bizonyult a legpontosabbnak (50 szerző esetén átlagosan 65.02%), így Ruder és társai is alátámasztották, hogy a karakterszintű konvolúciós modellek jobban teljesítenek azoknál a konvolúciós hálózatoknál, melyeknek bemenetei szóbeágyazások [32].

2.4.3. Szófajokon alapuló eljárások

Az előző fejezetben olyan konvolúciós hálózatokat mutattam be, mely a szöveget karakterek formájában kapják bemenetül. Ebben a fejezetben ismertetek olyan megoldásokat, ahol a szövegből kinyert szófajok lesznek az eljárások bemenetei. Két eljárást mutatok be, mely ilyen jellegű adatokon végzi a szerzős-megállapítást.

Hitschler és társai, 2017: Szerzőség-megállapítás konvolúciós neurális hálózatokkal és szófajegyesítéssel

Hitschler és társai olyan módszert javasoltak, ahol a neurális hálózat szószintű, bemenete pedig a szavak szófajai és beágyazásai egy bizonyos gyakoriság felett. Alapelgondolásuk az volt, hogy a szavak szófajait és a nyelvre jellemző leggyakoribb n szót együttesen használják fel egy konvolúciós hálózat bemeneti adataként.

Az eljárás felépítéséből és az adatok jellegéből elmondható, hogy a modell mintákat keres a szóbeágyazások koordinátái között, majd ezt követően a többi szóval együttesen fedezi fel az összefüggéseket. A modell azt ragadja meg, hogy a szerző a gyakrabban használt szavakat milyen mondatrészként használja, hogyan helyezi el azokat a mondatban, illetve mely szavakat használja. Például: ritkán előforduló szavakkal fejezi be a mondatot a szerző, az „azonban” szót a megelőzőekhez képest ellentétes kontextusú szavak követik-e, a „minden” szót a szerző jelzőként vagy alanyként használja, stb.

Az eljárás bemeneteinek elemei több részből tevődnek össze, melyeknek összefűzésével áll elő a végleges adat. Ezek a következők [34]:

- **Szófaj one-hot vektora:**

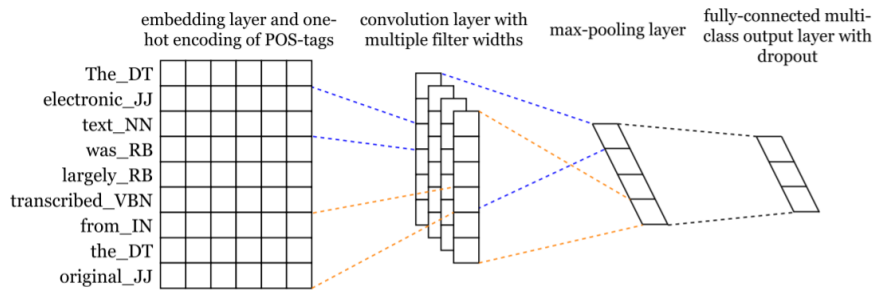
A szófajok kinyerésére a *Stanford* szófajcímkézőt alkalmazták, mely a *Penn Treebank* címkézésletével jelöli a szófajokat. A kinyert szófajok one-hot vektoros reprezentációját használják.

- **Szóbeágyazások:**

A bemeneti adatok a szavak szófaján kívül a szavak beágyazott formáját is tartalmazzák. A megközelítés alapgondolata, hogy a szófajon kívül jó leírónak tekinthető, hogy a szerző milyen módon használja a nyelvre jellemző n leggyakoribb szót. Ha a szó beleesik ebbe a csoportba, akkor annak statikus szóvektorát használja, egyébként pedig egy ismeretlen jelzésre utaló vektorral helyettesíti. A megvizsgált n küszöbértékek az 1 000, 5 000, 10 000, 50 000, „Nincs”. A szavak gyakoriságát egy független adathalmazon vizsgálták meg.

Kim által javasolt konvolúciós neurális hálózat mintájára tervezték a modellt, azzal a különbséggel, hogy többcsatornás hálózat helyett egy egyszerű, egycsatornás hálózatot valósítottak meg. A hálózat első rétege egy ún. *Embedding* réteg, melynek feladata csökkenteni az adat dimenzionalitását és azokat elhelyezni a vektortérben a mondatok kontextusát felhasználva. 100 darab 3 különböző méretű konvolúciós

szűrőt alkalmaztak, a teljesen kötött rétegeken 0,5 értékű dropout-ot alkalmaztak. A modell felépítése látható a 2.9 ábrán.



2.9. ábra. Hitschler és társai által javasolt konvolúciós modell felépítése[34]

A használt adathalmaz tudományos cikkeket tartalmaz a számítógépes nyelvészet területén 808 különböző szerzőtől. A szövegek témája kötött, ezzel biztosítva, hogy a módszer a tényleges szerzői stílust különböztesse meg, nem pedig a bemeneti szövegek témájára specifikus jellegzetességeket. Az adatok 1 500 szavas szegmensekre vannak felosztva, az ennél rövidebb szegmenseket eldobják.

A javasolt módszer 808 szerzőtől begyűjtött adatokon az $n = 50\,000$ szógyakoriság küszöbértékkel érte el a legjobb eredményeket. Meglepő módon, a teljes szótár megtartásával (az az eset, amikor $n = \text{„Nincs”}$) pontatlanabb eredmények érhetők el. Ennek ellenére elmondható, hogy az n értékének növelése jobb végeredménnyel jár a megvizsgált n értékekre. Az eljárás kiértékelését 14 szerzős *PAN 2012* adathalmazon is elvégezték az eredmények más módszerekkel való összehasonlíthatósága miatt. Ezen az adathalmazon az eljárás pontossága 78,57%.

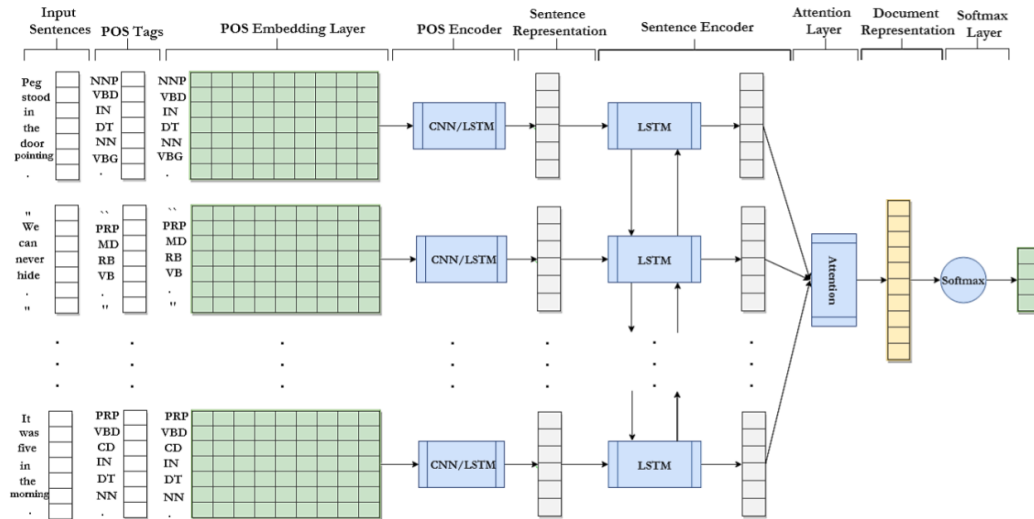
Jafariakinabad és társai, 2019: Szintaktikai rekurrens neurális hálózat szerzőség-megállapításra

Jafariakinabad és társai egy olyan modellt javasoltak, melynek bemenetei csakis a szavak szófajai. A modell a szófajokat mondatokba csoportosítva dolgozza fel. Ez a megközelítés Hitschler és társainak modelljétől jelentősen eltér, az ő modelljük jóval egyszerűbbnek jellemezhető. A szavak nincsenek mondatokba csoportosítva, a konvolúciós hálózat szavak sorozatában, a teljes szegmensen próbálja felfedezni a mintákat. Jafariakinabad és társainak modellje egy hierarchikus elemzést tesz lehetővé, az eljárás bemenetei a mondatba csoportosított szófajok melyekből kinyerhető a mondatrepresentáció. A további rétegekben attention mechanizmust alkalmazva előáll a dokumentumrepresentáció [33]. A modell felépítése látható a 2.10 ábrán.

A modell felépítése és a bemeneti adatok jellege arra utal, hogy az eljárás a szerző mondataalkotási szokásait ragadja meg. Például: az állítmánnyal kezdi a mondatot a szerző vagy esetleg más szófajú szóval, a szerző milyen szó után szokott vesszőket használni, tárgyak esetén szokott-e a szerző jelzőket használni, stb.

A szófajkódoló egység a szófajok beégyeztett reprezentációját kapják bemenetül. A szófajok ilyen jellegű reprezentációja sűrű vektorokat eredményez csökkentve a dimenzionalitást. Jafariakinabad és társai a módszerükben két megközelítést javasolnak a mondatrepresentáció kinyeréséhez: CNN és LSTM szófajkódoló használata, ahol az előbbi a rövidtávú, míg az utóbbi a hosszútávú függőségeket tanulja meg.

A szófajbeágyazások ezeknek a kódolóegységeknek a bemeneteit képezik. Az implementált rendszerben az NLTK csomag szófajcímkéző modulját alkalmazták, amely összesen 47 címkét használ. A hálózat mondatkódolásért felelős része egy kétirányú LSTM egység. A mondatkódoló egység kimenetei egy attention réteg bemeneteit képezi, feladata megjutalmazni a szerzői stílust jobban meghatározó mondatrepresentációkat. A hálózat utolsó rétege teljesen kötött, ahol a neuronok száma az osztályok számával egyezik meg.



2.10. ábra. Jafariakinabad és társai által javasolt modell felépítése [33]

A modell megtervezésekor szükséges egy maximális mondathossz meghatározása. A használt adathalmaz mondatainak átlagos hossza 20 szó. Maximális mondathossznak a 30 szót határozták meg, így nincs jelentős információvesztés a mondatok levágásakor, viszont nem lesz feleslegesen nagy a modell mérete. A bemeneti adatok hossza 100 mondat. A kiértékelésre használt adathalmaz a *PAN 2012* projekt 14-szerzőből álló adathalmaza, ahol az eljárás CNN-LSTM variációja 78,76%-os pontosságot ért el, míg az LSTM-LSTM változat 74,40%.

A két szófajon alapuló modellről elmondható, hogy hasonló pontosságot érnek el egy 14-szerzős adathalmazon. Hitschler és társainak modellje egyszerűbb felépítésű, azonban az adatok előkészítése bonyolultabb, így szófajcímkézőn kívül szükség van előre tanított szövektorokra is. Jafariakinabad és társainak módszere ezzel szemben kevesebb külső eszközt igényel, ugyanis a modell bemenetei a szavak szófajai mondatokba csoportosítva. Modelljük egyik hátránya, hogy összetett és nagy méretű.

2.4.4. Szóbeágyazásokon alapuló eljárások

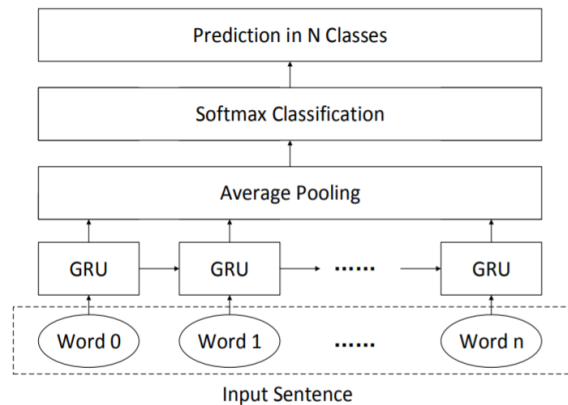
Az előző két fejezetben olyan eljárásokat ismertettem, melyek karaktereket vagy szófajokat kapnak bemenetül, ebben a fejezetben részletezem egy olyan módszer működési elvét, ami a szavak szóbeágyazásait használja fel.

Chen és társai, 2017: Mélytanuláson alapuló szerzőség-elemzés

Chen és társai olyan módszereket javasolnak, mely a szöveget idősoros adatként kezeli, így LSTM és GRU hálózatokat terveztek meg a szerzőség-elemzés probléma-

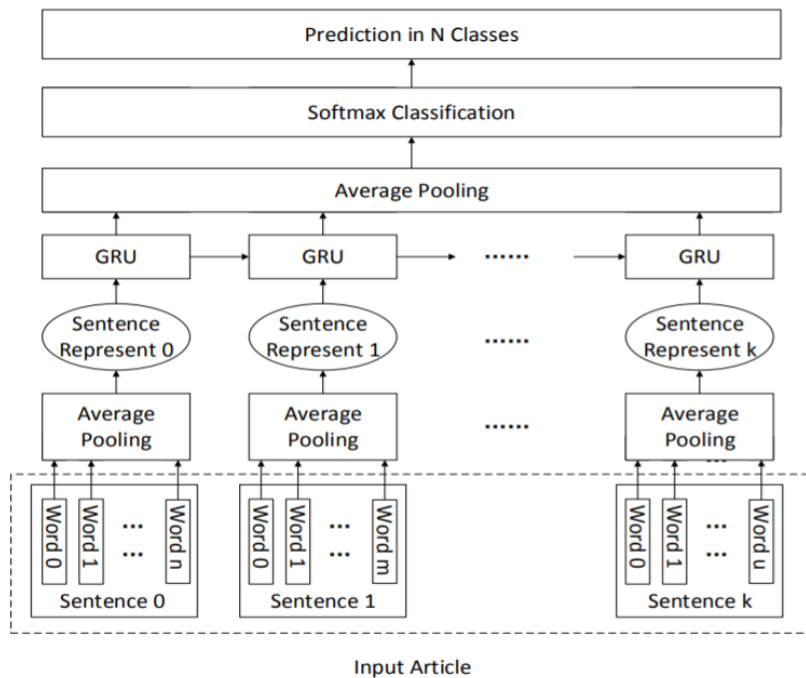
körére. A két javasolt módszer különböző méretű adatokat kap bemenetül, az egyik a statikus szövektorokat, míg a másik a mondat egy olyan reprezentációját, ami a mondat szövektorainak súlypontjának felel meg [35].

A mondat szintű GRU vagy LSTM modellek bemeneti adata kisszámú (általában) 3 egymást követő mondat szóbeágyazásai. A GRU vagy LSTM egységek kimenetei egy összevonó (*average-pooling*) réteg bemenetére kerülnek, melynek kimenete a az egyes értékek átlaga. A modell utolsó része a softmax aktivációs függvény, ami előállítja a végleges döntést. A modell a 2.11 ábrán látható.



2.11. ábra. Chen és társainak által javasolt mondat szintű modell felépítése [35]

A cikks szintű GRU vagy LSTM modellek abban különböznek a mondat szintű modellektől, hogy nem az egyes szóbeágyazások értékei képezik a GRU vagy LSTM egységek bemeneteit, hanem azok átlagos értéke, ami vektorok esetén a súlypontnak felel meg. A modell a 2.12 ábrán látható.



2.12. ábra. Chen és társai által javasolt cikks szintű modell felépítése [35]

Az eljárást két különböző adathalmazon értékelték ki:

- **C50 adathalmaz:** Az adathalmaz 50 szerző által megírt vállalati vagy iparí témájú híreket tartalmaz. Az adathalmaz mérete 50 000 (100 szerzőnként), ahol az adatokat 9:1 arányban használták fel tanításra és tesztelésre.
- **Gutenberg adathalmaz:** Az adathalmaz szerteágazó témájú és szerzőjű elektronikus könyvet tartalmaz. A szövegeket bekezdésekre osztották fel, ilyen módon szerzőnként 900, összesen 45 000 bekezdésből tevődik össze az adathalmaz. Az adatokat szintén 9:1 arányban osztottak fel tanításra és tesztelésre.

A mondatszintű GRU vagy LSTM hálózatok nem bizonyultak hatékonyak a szerzőség-elemzés problémájára, a modell túltanult a kevés bemeneti adat és a sok paraméter következtében.

A cikkszintű GRU hálózatok a *C50* adathalmazon 69,1%-os, míg a *Gutenberg* adathalmazon 89,2%-os pontosságot eredményezett. A GRU egységekből álló hálózat jobb eredményeket biztosított az LSTM hálózathoz képest. A szerzők megemlítik publikációjukban, hogy a *Gutenberg* adathalmaz szerzőinek stílusa nagyon egyedi, így a *C50* adathalmazon kiértékelt eredményeket tekintik reprezentatívnak.

2.4.5. Megvizsgált eljárások összehasonlítása

A megvizsgált eljárások más-más architektúrákon és adattípusokon végzik a szerzőség-megállapítást. Zhang és társai karakterszintű konvolúciós hálózatot javasoltak általános szövegklasszifikációs feladatokra. Az eljárás nagyon nagy adathalmazon és rövid, 1014 karakterű szöveges adatok esetén biztosítja a legpontosabb eredményt. A 2.3 táblázat első sorában látható egy általános összegzése az eljárásnak. Az eljárás kiértékelése során használt adathalmazok egyike van feltüntetve a táblázatban. Ezeken az adatokon az elért pontosság 72,2% [31]. Ruder és társai szerzőség-megállapításra alkalmazták az eljárást, és megerősítették a rendszer pontosságát ebben a problémakörben is [32]. Érdekes megjegyezni, hogy a Zhang és társai által a karakterek one-hot vektorok formájában kódoltak, érdemes lehetne a betűk beágyazott formáit használni, ugyanis a betűk nem tekinthetők független kategórikus adatnak (például a magánhangzók és mássalhangzók általában váltakoznak a szavakban).

Hitschler és társai egy olyan módszert javasoltak, melynek bemenetei a szavak beágyazásainak és szófajok one-hot reprezentációinak összefűzött változatai. Az eljárás során használt modell szintén egy konvolúciós neurális hálózat, amelynek bemeneti adatainak hossza 1500 szó. Ahogyan az a 2.3 táblázatban látható, a teljes adathalmaz mérete 1894 szöveg, ami jelentősen kisebb számú adatot jelent az első eljáráshoz képest. Az elért pontosság 78,57% [34].

Jafariakinabad és társai Hitschler és társaihoz hasonlóan a kinyert szófajokat használják fel szerzőség-megállapításra. Az általuk javasolt modell hierarchikus elemzést végez, ahol a szófajreprezentációk között keres összefüggéseket az eljárás ezzel megtanulva a szerzőre jellemző mondatokstruktúrákat. A mondatreprezentációkat egy attention réteggel súlyozzák, így kialakul egy dokumentumreprezentáció [33]. Az eljárást 100 mondat hosszú szövegszegmenseken értékelték ki 100-150 adaton szerzőnként. Jafariakinabad, Hitschler és társai ugyan azt a 14-szerzős adathalmazt használták, így az eredményeik közvetlenül összehasonlíthatók, jelentős eltérés a két módszer pontossága között nem figyelhető meg, ahogyan az a 2.3 táblázatban is látható.

Chen és társai által javasolt eljárás az egyes szövegszegmensek szóbeágyazásait kapja bemenetül mondatokba csoportosítva. A mondaton belüli szóbeágyazások átlaga képezi a GRU vagy LSTM egységek bemeneteit. Az eljárást egy 50-szerzős 50 000 adatból álló adathalmazon értékelték ki néhány bekezdés terjedelmű szövegszegmenseken, melyeknek pontos hossza nincs feltüntetve a publikációban. Az adathalmazon elért pontosság 69.1% [35].

Eljárás	Bemeneti adat típusa	Architektúra típusa	Bemeneti adat hossza	Adathalmaz mérete	Osztályok száma	Pontosság
Zhang és társai	Karakterek	CNN	1014 karakter	1 460 000	10	72,2%
Hitschler és társai	Szófajok, szóbeágyazások	CNN	1500 szó	1894	14	78,57%
Jafariakinabad és társai	Szófajok	CNN, LSTM	100 mondat	100-150 / szerző	14	78,76%
Chen és társai	Szóbeágyazások	GRU / LSTM	Néhány bekezdés	2 500 / szerző	50	69,1%

2.3. táblázat. Megvizsgált eljárások összefoglaló táblázata

Az eljárások több szempont szerint hasonlíthatók össze:

- **Bemeneti adat típusa és kinyerésükhöz szükséges külső eszközök:**
A karakterszintű megoldások nagy előnye, hogy nyelvfüggetlenek. Egy nyelvre specifikus abécén kívül másra nincs szükség az adat kinyerésére. Ezzel szemben a szófajok kinyeréséhez szükség van szófajcímkéző eszközökre, míg szóbeágyazások felhasználásakor előre tanított modelleket szokás alkalmazni a kezdeti súlyok beállításához.
- **Architektúra típusa, hálózat tanítása:**
Két típusú architektúra jelenik meg a megvizsgált eljárások között, konvolúciós és LSTM vagy GRU hálózatok. A fő különbség a két típus között az, hogy a konvolúciós hálózatok nem idősoros adatként kezelik az adatokat, míg az LSTM és GRU hálózatok igen. Egy másik nem elhanyagolható szempont a műveletek párhuzamosíthatósága a két architektúrában. A konvolúció egyszerű, független algebrai műveletekből áll. Az LSTM és GRU hálózatok szűk keresztmetszete az adatok időbelisége, így az egységek műveleteinek nagy része szekvenciális végrehajtást igényel. Ennek következménye, hogy a GRU és LSTM hálózatok tanítása sokkal lassabb grafikus kártyán a konvolúciós hálózatokhoz képest.
- **Bemeneti adat hossza:**
A karakterszintű megoldások pár mondat hosszúságú szövegszegmenseken teljesítenek a legjobban. A szófajokat általában hosszabb bemeneti szövegeken alkalmazzák, például 100 mondat hosszúságú szegmenseken. A szóbeágyazásokat Chen és társai néhány bekezdés hosszú adatok kiértékelésére használják.
- **Adathalmaz mérete:**
Bizonyos eljárások eredményessége pont a nagy adathalmaz használatában rejlik, így azok kevés rendelkezésre álló szöveg esetén nem biztosítják az ígért pontosságot. Egy ilyen eljárás a Zhang és társai által javasolt módszer. Hitschler, Jafariakinabad és társaik módszereiket a másik kettő megvizsgált eljáráshoz képest jóval kisebb adathalmazon vizsgálták meg, ennek ellenére az eljárások pontossága nem különbözik nagymértékben.
- **Pontosság:**
A megvizsgált eljárások eredményessége függ az adathalmazon elért pontos-

ságtól, ezek azonban nem összehasonlíthatók egymással egyéb szempontok figyelembevételével. A pontosság nagyban függ attól, hogy hány szerzős adathalmazon végezték a kiértékelést, így például Chen és társai 69,1%-os pontossága nem tekinthető rosszabb eredménynek, mint Jafariakinabad és társainak vagy Hitschler és társainak 79%-os pontossága, ugyanis az osztályok száma eltér. Több osztály esetén az eljárás pontossága csökken. Emellett, fontos figyelembe venni az adathalmaz témáját, jellegét. Chen és társai kiértékelték az eljárást híreket és irodalmi jellegű szövegeket tartalmazó adathalmazokon, ahol az utóbbi esetén jóval nagyobb pontosságot értek el az irodalmi szövegek megfigyelhető egyedibb szerzői stílus következtében.

A megvizsgált eljárásokról elmondható, hogy a bemeneti adatoktól függ a számunkra megfelelő rendszer megválasztása. Ha nem szeretnénk külső eszközöket használni a szöveg feldolgozására vagy a szóbeágyazások tanítására, akkor Zhang és társainak eljárása javasolt. Emellett elmondható, hogy Zhang és társainak modellje nagyon egyszerűen implementálható és adaptálható magyar nyelvű szövegekre, azonban nagyon nagy adathalmazokon eredményez jobb osztályozást más rendszerekhez képest. Chen és társai által javasolt modell felépítése szintén nagyon egyszerű, és könnyen implementálható előre tanított szövektorok rendelkezésre állása esetén. Jafariakinabad és Hitschler és társainak eljárásához szintaktikai szövegelemzésre van szükség. Hitschler és társainak modellje egyszerű felépítésű, azonban az adatok előkészítése bonyolultabb a többi ismertetett eljáráshoz képest, ugyanis szófajokon kívül szükség van a szavak statikus szövektoraira is. Jafariakinabad és társainak modellje bonyolultabb, viszont az adatok előkészítése egy szófajcímkéző megléte esetén egyszerűnek tekinthető.

3. fejezet

Eljárások kiértékelése magyar nyelvű szövegeken

Az előző fejezetben bemutatam néhány szerzőség-megállapításra javasolt mély neurális hálózatokon alapuló eljárás működési elvét. A módszerek mindegyike angol nyelvű szövegeken kerültek kiértékelésre. Kutatási célom az eljárások vizsgálata magyar nyelvű szövegeken, így ebben a fejezetben bemutatom az irodalomkutatás során megvizsgált valamennyi módszer magyar nyelvre adaptált változatát, illetve a magyar nyelvű adatokon elért eredményeket.

3.1. Adathalmaz

Léteznek publikus magyar nyelvű korpuszok, azonban olyan adathalmazok, amelyek alkalmasak lennének szerzőség-megállapításra készült modellek tanítására nem állnak rendelkezésre. Emiatt, egy általam összegyűjtött és címkézett adathalmazt használtam.

A dolgozatom korábbi fázisában szintén egy saját adathalmazon értékeltem ki az eljárásokat, azonban manuálisan történt az adatok begyűjtése a MEK (Magyar Elektronikus Könyvtár) [38] weboldaláról, tisztítása, címkézése. A begyűjtött adathalmaz kicsinek bizonyult, összesen 14,5 MB adatot gyűjtöttem be 23 szerzőtől (0,63 MB szerzőnként), ami az adatok kiegyensúlyozását követően még tovább csökkent. Abban az esetben ha szerzőnként eltérő mennyiségű adat áll rendelkezésünkre, a legkevesebb adatot tartalmazó szerző szövegrészleteihez kell igazítani a szerzőnkénti szövegmennyiséget, ezzel elkerülve, hogy a hálózat a több szövegrészlettel rendelkező szerzőket favorizálja, ezt értem adatok kiegyensúlyozása alatt. Például a korábbi adathalmaz esetén 100-mondatos szövegrészletekre bontva az adatokat, szerzőnként 34 szövegrészlet jutott. Tekintve, hogy olyan eljárásokat is megvalósítok a dolgozatomban, melyek kizárólag nagyobb mennyiségű adaton teljesítenek jól, nélkülözhetetlennek éreztem egy nagyobb korpusz begyűjtését.

Szükségesnek éreztem az adatok automatikus begyűjtését, tisztítását és címkézését, ugyanis az eredményül kapott adathalmaz ez esetben könnyen skálázható. Akár más műfajú korpusz is minimális erőbefektetéssel tölthető le ezáltal, nincs szükség emberi erőforrásra az adatok címkézéséhez. A programot java programozási nyelvben valósítottam meg a *Selenium* keretrendszerben. A keretrendszer weboldalak tesztelésére készült, webdrivereket segítségével a HTML DOM elemeinek selectorjait definiálva könnyen azonosíthatunk gombokat, mezőket, olyan műveletek is elvégez-

hetők vele mint például a gépelés vagy a kattintás. A keretrendszer használata közel áll a tényleges felhasználói élményhez, ugyanis a böngésző ablaka is látható a műveletek futása közben.

1. Algorithm Kórpusz letöltésének folyamata a MEK honlapjáról

```

1: function DOWNLOADDOCUMENTS
2:   documents  $\leftarrow$  []
3:   authors  $\leftarrow$  GETAUTHORSLIST()
4:   for author in authors do
5:     documentTiles  $\leftarrow$  SEARCHAUTHOR(author)
6:     for documentTile in documentTiles do
7:       title  $\leftarrow$  documentTile.title
8:       if ISVALIDTITLE(title) then
9:         documentTile.click()
10:        htmlLink  $\leftarrow$  FINDHTMLLINK()
11:        if htmlLink  $\neq$  null then
12:          htmlLink.click()
13:          documentText  $\leftarrow$  READANDFILTERHTMLTEXT()
14:          if  $\neg$ ISDRAMA(documentText) then
15:            document  $\leftarrow$  new()
16:            document.text  $\leftarrow$  FILTERLINES(documentText)
17:            document.author  $\leftarrow$  author
18:            documents.ADD(document)
19:          end if
20:        end if
21:      end if
22:    end for
23:  end for
24:  return documents
25: end function

```

Az algoritmus részletes működését az 1 algoritmus írja le. A használt segédfüggvények a következők:

- **GETAUTHORSLIST**

A függvény a Wikipédián megtalálható magyar költők és írók listájából [39] gyűjti a magyar szerzők neveit. Csak azokat a szerzőket gyűjti ki a program, melynek leírásában szerepel az „író” szó, ezzel kiszűrve azokat a szerzőket, melyek kizárólag versírással, műfordítással, stb. foglalkoznak. A dolgozatomban prózai, eredeti műveken végzem a szerzőségelemzést, így fontosnak tartom a műfordítások, versek, drámák kiszűrését, ennek folyamatát a továbbiakban ismertetem.

- **SEARCHAUTHOR**

A függvény a böngészőablakban elnavigál a Magyar Elektronikus Könyvtár összetett szűrőfelületére, a paraméterül megadott szerző nevét beírja a keresőmezőbe, majd rakattint a keresés gombra. A keresés eredményét ún. „csempéknek” (*documentTile*) nevezem. A csempéken a művek címei $\{szerző1\}, \{szerző2\}, \dots : \{Mű címe\}$ formátumban szerepelnek, a csempe alsó részén pedig a mű url-je található meg.

- **IsValidTitle**

Mint ahogyan az előző pontban megemlítettem, a címnek kötött formátuma van: $\{szerző1\}, \{szerző2\}, \dots : \{Mű\} címe$. A cím ennek függvényében többféleképpen állhat elő:

1. A műnek nincs szerzője, csak címe (nincs ':' karakter a címbe). Ez általában enciklopédiák, szöveggyűjtemények esetén fordul elő.
2. A műnek egy szerzője van (a ':' előtt nem szerepel vessző).
3. A műnek több szerzője van (a ':' előtt szerepel vessző)

A felsorolt esetek közül a második esetén tér vissza a függvény igazgal, egyébként hamissal, ugyanis szerzőség-elemzésre alkalmas korpusz begyűjtése esetén rendkívül fontos a többszerzős művek kiszűrése.

- **FindHTMLLink**

A függvény megnézi, hogy elérhető-e az aktuális dokumentum HTML formátumban. Amennyiben igen, kigyűjti az url-t amin elérhető, egyéb esetben pedig üres értékkel tér vissza.

- **ReadAndFilterHTMLText**

A függvény kiolvassa azoknak a bekezdéseknek a szövegét ($\langle p \rangle$ elemek, vagy $\langle p \rangle$ elemeken belül a $\langle font \rangle$ típusú elemek), melyek sorkizártan vannak igazítva és a betűvastagságuk normál. A feltételnek eleget tevő bekezdések szövegeinek összefűzött formájával tér vissza az algoritmus. A három feltétellel a következőket szűrjük ki:

- A **versek** általában vagy középre, vagy balra vannak rendezve, de soha nem sorkizártak, így a versrészeket nem fogja kiemelni az algoritmus.
- **Cím, szerző neve, kiadó neve, egyéb metainformáció** általában más stílusú szöveggel szerepel a mű szövegéhez képest. Általában ezek nagyobb font mérettel, vastagon szedett betűkkel jelennek meg, így az algoritmus ezeket is kiszűri.
- A **tartalomjegyzékben** szereplő elemek többnyire $\langle a \rangle$ elemeken belül találhatóak meg és vastagon szedettek. Azáltal, hogy az algoritmus a $\langle p \rangle$ elemeken belül keresi a szöveget, kiszűri a tartalomjegyzéket is.
- Az **oldalszámok és fejezetcímek** középre vannak igazítva, így azok sem fognak belekerülni a kigyűjtött szövegbe.

- **IsDrama**

A függvény a paraméterül megkapott szöveg alapján dönti el, hogy az dráma-e vagy sem. Ezt oly módon teszi meg az algoritmus, hogy megvizsgálja a bekezdés első néhány betűjét. Megszámolja hány olyan bekezdés van összesen, melynek első k betűje mind nagybetűs karakter, legyen ez az érték egyenlő p -vel. Amennyiben $p \cdot t > n$, akkor a szöveg drámának számít (n a bekezdések száma és t egy kiszűbörték). Az algoritmusban a $t = 15$, $k = 4$ értéket használtam, ami azt jelenti, hogy ha csak minden 15. sornak első 4 betűje nagybetűs, az még nem számít drámának, viszont ha ez több bekezdésre teljesül, akkor már igen.

- **FILTERLINES**

A függvény kiszűri a számokkal kezdődő és üres sorokat. A sorok általában akkor kezdődnek számokkal, ha a mű végén fogalomtár vagy hivatkozásjegyzék szerepel.

Az algoritmus első lépése meghatározni azoknak a szerzőknek a nevét, melyeknek műveit szeretnénk kigyűjteni, ezt a Magyar szerzők listája Wikipédia cikkből [39] határoztam meg. Ezt követően az algoritmus az összes szerző nevét felhasználva összetett keresést végez. Az eredményül kapott művek listájából kiszűri a többszerzős műveket, majd kiolvassa a mű szöveges tartalmát, különös tekintettel arra, hogy a zajkeltő szövegrészeket, mint például szerző és mű címe, tartalomjegyzék, oldalszámok, hivatkozásjegyzék, stb. kiszűrje. Emellett, a szövegen végzett összetettebb elemzés is nélkülözhetetlen a drámák kiszűrése érdekében. Az algoritmus kimenete a szépirodalmi prózai művek szöveges formája a szerző nevével együtt. Az adatokat a program a háttértárra szöveges állományok formájában menti le, ahol az egyes műveket `{szerző}_{mű címe}.txt` konvencióval nevezi el, így tehát az annotáció maga az állomány neve a ' _ ' karakterig.

A korpusz összesen 183 szerzőtől begyűjtött műveket tartalmaz, mérete 155 MB (0,85 MB szerzőnként). A teljes korpusz használata az eljárások kiértékelésére nem célszerű, ugyanis bizonyos szerzőktől nagyon csekély mennyiségű adatot tudtam begyűjteni. Abban az esetben, ha a teljes korpuszt fel szeretném használni és az eljárásokat kiegyensúlyozott adatokon kiértékelni, a begyűjtött adatok nagy részét el kell távolítanom. Emiatt, a teljes korpusz töredékén végzem a kiértékelést, ami 15 szerző műveiből áll. A korpusz azon szerzők szövegeiből áll, melyektől a legtöbb adatot gyűjtöttem be. A csökkentett méretű korpusz mérete 58 MB (3,85 MB szerzőnként). Az adatok kiegyensúlyozását követően ez összesen 199 100-mondatos szövegrészletet jelent szerzőnként, ami jelentősen több a korábbi korpusz méretéhez képest (34 100-mondatos szövegrészlet).

3.2. Tanítás körülményei

A modellek python programnyelvben vannak megvalósítva, *Tensorflow* és *Keras* keretrendszerek segítségével. Angol nyelvre implementált forráskódok általam módosított változatát alkalmaztam. A modellek tanítását részben a Google Colaboratory virtuális környezetben végeztem, részben pedig egy virtuális számítógépen grafikus kártyán.

A modellek tanítását eltérő számú iteráción keresztül végeztem. A tanítás során *early stopping callback*-et alkalmaztam *patience* = 4 értékkel a validációs veszteségre nézve. Ez azt jelenti, hogy a tanítás leáll, amikor a legalacsonyabb validációs veszteség nem csökken 4 iteráción keresztül. A tanításra vonatkozó adatokat grafikonok formájában ismertetem, ahol a késsel jelölt vonal a validációs pontosságot, a szaggatott kék vonal a tanítási pontosságot, a piros vonal a validációs veszteséget, míg a piros szaggatott vonal a tanítási veszteséget jelöli. A pontosságokat minden esetben [0,1] intervallumon ábrázolom, míg a veszteségeket relatív módon. Ennek oka, hogy a kezdeti veszteség modelltől függően fog változni, viszont az értékek ennek ellenére jó betekintést adnak arra, hogy mennyire tanul gyorsan a hálózat. A vízszintes tengelyen az iterációk száma látható. A kiértékelt modellekre pontosságot számszerűsítve is megállapítok, a továbbiakban pontosság alatt azt a validációs pontosságot értem,

ahol a validációs veszteség a legalacsonyabb.

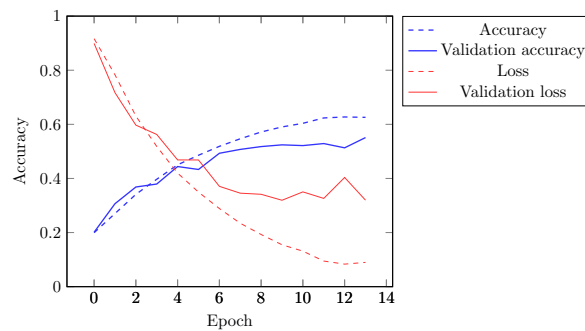
3.3. Kiértékelt eljárások

Ebben a fejezetben részletezem, mely módszereket vizsgáltam meg magyar adatokon, illetve milyen lépésekkel egészítettem ki az eredeti eljárást, hogy az magyar nyelvű szövegek szerzőség-megállapítására is alkalmas legyen.

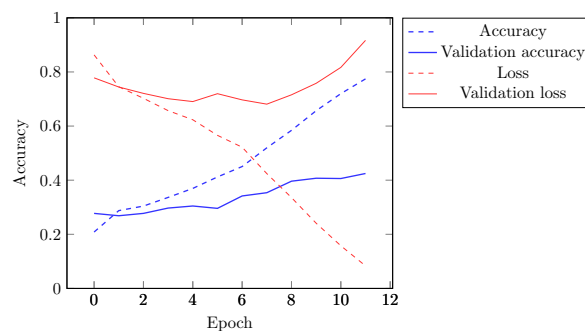
3.3.1. Karakterszintű konvolúciós neurális hálózat

Zhang és társainak eljárását vizsgáltam meg és adaptáltam magyar nyelvre [31] a [40] forráskód módosított változatával. Mint ahogy azt az irodalomkutatás fejezetben is részleteztem, Zhang és társainak módszerének nagyon nagy előnye, hogy egyszerű és nyelvfüggetlen. A modellnek két variációját ismertették. Az egyik változat konvolúciós rétegeinek mérete 256 és a másiké pedig 1024 [31], én az előbbit alkalmaztam a méréseim során.

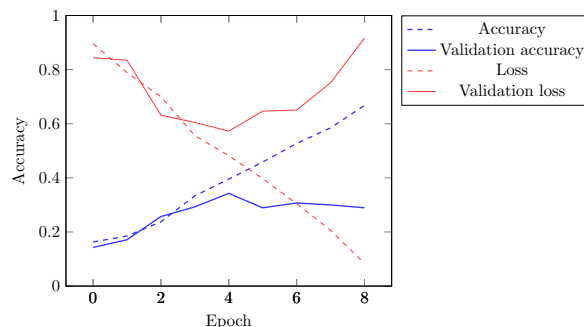
Az adatok beolvasását követően kiszűröm a különleges karaktereket, amelyek zajként hatnak az adatra. Az angol és magyar ábécé közötti jelentős nagy különbség, hogy az utóbbi tartalmaz két- és háromkarakteres betűket. Annak érdekében, hogy az összes karakter egy betűt reprezentáljon, a magyar többkarakteres betűket különleges karakterekkel helyettesíttem. Az adatok N mondat hosszúságú szegmensekre bontása után alkalmazható magyar nyelvű szövegekre az eljárás. A modell kiértékelésére, tanítására vonatkozó adatok láthatók a 3.1, 3.2 és 3.3 ábrákon különböző N mondat hosszúságú szövegszegmenseken.



3.1. ábra. Zhang és társainak modelljének tanítása 5 mondat hosszúságú adatokon



3.2. ábra. Zhang és társainak modelljének tanítása 30 mondat hosszúságú adatokon



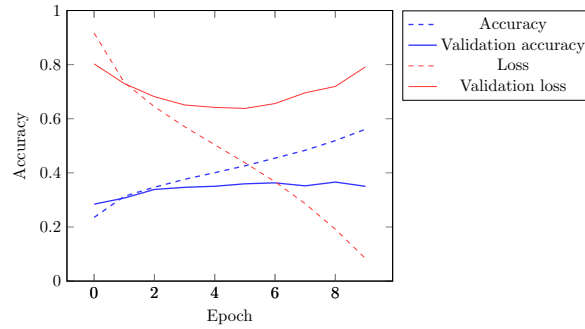
3.3. ábra. Zhang és társainak modelljének tanítása 100 mondat hosszúságú adatokon

A 3.1, 3.2, 3.3 ábrákat összehasonlítva, azt láthatjuk, hogy az eljárás 5 mondat hosszúságú adaton tanul a legtovább, illetve ilyen adatok esetén éri el a legnagyobb pontosságot (5-mondathosszúságú adatokon 52,42%, 30-mondathosszak esetén 35,37% és 100-mondathossz esetén 34,28%). A szövegrészletek növelésével egyre rosszabban teljesít a hálózat, így beláthatjuk, hogy az eljárás a legjobban rövidebb szövegrészletek esetén végzi a klasszifikációt 15 osztályra.

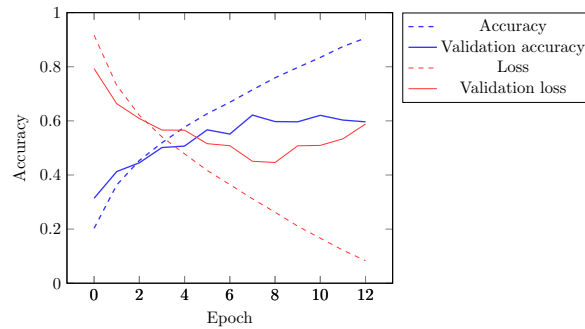
3.3.2. Szintaktikai rekurrens neurális hálózat

Zhang és társainak eljárásán kívül Jafariakinabad és társainak [33] módszerét is megvizsgáltam szerzőség-megállapításra az általam begyűjtött adathalmazom a [41] forráskód általam módosított változatán. Jafariakinabad és társainak eljárásának adaptációja magyar nyelvre az előző rendszerhez képest összetettebb és több lépésből áll.

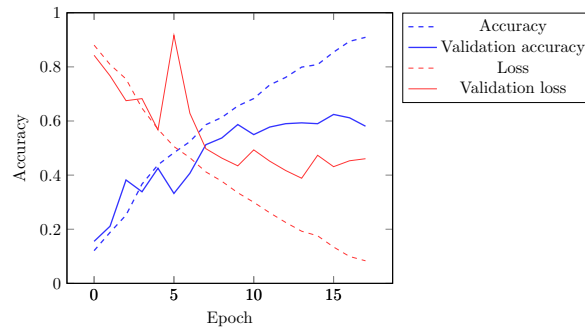
A szófajok kinyerésére Jafariakinabad és társai az NLTK csomag szófajcímkéző modulját használják, mely összesen 47-féle értéket rendelhet az egyes szavakhoz, ugyanis az eszköztár a *Penn Treebank* címkekeszletét alkalmazza. Ezzel szemben a magyar nyelvre elérhető szófajcímkézők, mint amilyen a magyarlanc [15] ennél kisebb méretű címkekeszlettel rendelkeznek. A magyar nyelv morfológiailag gazdag nyelvnek számít, így a szavak szintaktikai szerepe nem jellemezhető 47-féle értékkel. Emiatt, az UD címkekeszletet használja a magyarlanc eszköztár, melynek összesen 14 féle értéke lehet, kiegészítve a szófajra vonatkozó egyéb opcionális morfológiai jellemzőkkel (2.1 táblázat 4. oszlopában a szófajok, 5. oszlopában az egyéb morfológiai jellemzők láthatók). Ezek alapján belátható, hogy Jafariakinabad és társainak módszere nem adaptálható teljes mértékben magyar nyelvű szövegekre. Az egyéb morfológiai jellemzők figyelembevételével a bemeneti adatok értékkészlete nagyon megnő, így ehelyett a szó függőségi címkéjét használom fel, mint kiegészítő adat. Tehát, Jafariakinabad és társai módszerének magyar nyelvre adaptált változatában a bemeneti adat a szófaj helyett a szófaj és a függőségi címke párosa. A modell kiértékelésére, tanítására vonatkozó adatok láthatók a 3.4, 3.5 és 3.6 ábrákon különböző N mondat hosszúságú szövegszegmenseken.



3.4. ábra. Jafariakinabad és társainak modelljének tanítása 5 mondat hosszúságú adatokon



3.5. ábra. Jafariakinabad és társainak modelljének tanítása 30 mondat hosszúságú adatokon



3.6. ábra. Jafariakinabad és társainak modelljének tanítása 100 mondat hosszúságú adatokon

A 3.4, 3.5, 3.6 ábrákat összehasonlítva, azt láthatjuk, hogy az eljárás a legjobban a hosszabb szövegeken teljesít. Az iterációk száma a hosszabb szövegrészletek esetén nagyobb, azonban az elért pontosság hasonló érték körül mozog. A legkisebb szövegrészleteken teljesít a legrosszabbul az eljárás, ahol az látható, hogy a hálózat kevés ideig tanul és nem éri el a másik két esetben elért pontosságot (5-mondathosszúságú adatokon 35,95%, 30-mondathosszak esetén 59,74% és 100-mondathossz esetén 59,31%).

3.3.3. Szerzőség-megállapítás konvolúciós neurális hálózatokkal és szófajjegyesítéssel

Hitschler és társainak eljárásának működését is megvizsgáltam magyar szövegeken. A modell felépítése nagymértékben hasonlít Kim [30] által javasolt modellre, így

annak egy implementációját [42] használtam fel az eljárás kialakítására.

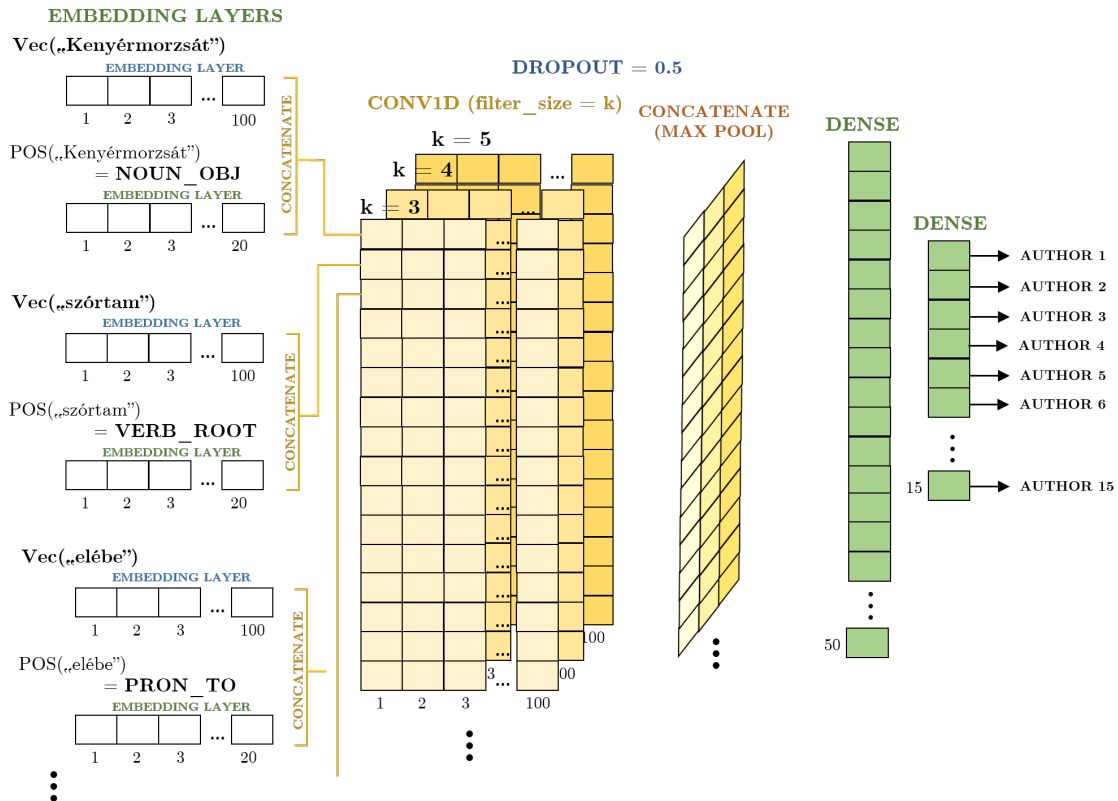
Bemeneti adatok

Az előző eljáráshoz hasonlóan, Hitschler és társainak módszere felhasználja a szavak szófajait. A szófajok kinyeréséhez szintén a magyarlanc eszköztárat alkalmaztam, valamint a szófajokat ismét a mondatban betöltött szerepükkel egészítem ki a gazdagabb címkekészlet elérése érdekében.

Mint ahogyan azt a Szóbeágyazások című fejezetben megemlítem, a szépirodalmi szövegeken végzett szerzőség-azonosításra kiválóan alkalmazható a *transfer-learning*. A *Szeged word vectors* fastText szóbeágyazásokat használok fel a szépirodalmi szóbeágyazások súlyainak inicializálására. Tekintettel arra, hogy a *Szeged word vectors* 100-dimenziós szövektorokat tartalmaz, a tovább tanított szóbeágyazások dimenziója szintén 100. A szóbeágyazásokat oly módon tanítom tovább, hogy egy *Embedding* réteg bemenetére kapcsolom rá a szavakat, mely a feldolgozott szövegre fogja adaptálni a szóbeágyazások súlyait.

Az eredeti eljárás szövegrészleteinek hosszát szavak száma, nem pedig mondatok száma határozza meg. Annak érdekében, hogy az eljárás teljesítménye összehasonlítható legyen a másik két kiértékelt eljárással, egy mondatot 30 szó sorozatának veszek. Így tehát az 5 mondathosszúságú szegmensek 150, a 30 hosszú szegmensek 900, míg a 100 hosszú szegmensek 3000 szóból állnak.

Modell felépítése



3.7. ábra. Hitschler és társainak általam adaptált eljárása

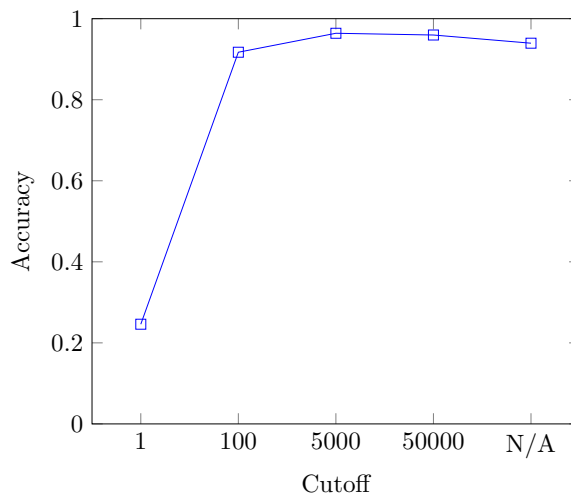
Hitschler és társai által javasolt eljárás bemenetül a szavak statikus szövektorait és a szófajok one-hot vektorait kapja. A bemenet dimenzionalitását csökkentve és azok vektortérben való megfelelő elhelyezése érdekében egy ún. *Embedding* réteggel tanítja tovább. Az implementációmban ettől eltérő módon jártam el, ugyanis véleményem szerint a szófajokat és a szóbeágyazásokat külön-külön vektortérben reprezentatívabban el lehet helyezni, így tehát a hálózatomban két külön bemenete van: statikus szóbeágyazások és a szófajok. Ezeket külön-külön tanítom két *Embedding* réteggel, ahogyan az az 3.7 ábrán látható. A szófajok *Embedding* rétege 20-dimenziós vektorokat eredményez, míg a szóbeágyazások dimenziója 100. A kapott vektorokat összfűzöm, ez a vektor képezi a konvolúciós réteg bemenetét. Itt 3 különböző k méretű szűrővel dolgozik a konvolúciós réteg ($k = 3, k = 4, k = 5$). A konvolúciós rétegeken *dropout* mechanizmust alkalmaz az algoritmus, mely egy regularizációs eljárás. Az aktivációs térképek alkalmazását követően a 3 konvolúciós rétegből összevonó (*max-pooling*) eljárással a vektorok mindegyikéből kiválasztja a legnagyobb értékű koordinátát, majd a 3 értéket összefűzi egy vektorba. A kapott vektorok képezik a teljesen kapcsolt rétegek bemenetét, az implementációmban 50 neuronból álló réteget használok. A modell utolsó rétege a kimeneti réteg mely szintén teljesen kapcsolt.

Vágás (*cutoff*) értéke

Hitschler és társai az eljárásukban a szótár leggyakoribb n szó beágyazott formáit használják fel, a kevésbé gyakori szavakat pedig egy ismeretlen jelzésű vektorral helyettesítik. Az implementációm során a következő n értékekre vizsgáltam meg az eljárást: $n = 1, 100, 5\ 000, 50\ 000, N/A$. A leggyakoribb n szót a tanítási adathalmazból határozom meg.

Az eljárás szóbeágyazásokat használ annak érdekében, hogy megragadja melyik szerző milyen módon használ egyes szavakat. Abban az esetben, ha egyes szerzők egy meghatározott témát favorizálnak, előfordulhat, hogy a hálózat a tematikára jellemző szavakat tanulja meg a tényleges szerzői stílus helyett. Például, ha egy szerző a korpuszunkban szereplő művei közül mind szerelmes regény, vélhetően az összes szövegben sokszor fog megjelenni a „szerelem”, „szeret”, „szerelmes”, „szem”, „ajak”, „szép”, stb. szavak.

A témakörre jellemző szavak megtanulása oly módon kerülhető el, hogy csökkentjük az n vágás értékét. Ez általában a pontosság csökkenésével jár, viszont kisebb lesz a valószínűsége, hogy egy, a hálózat által ismert szerzőjű de más témájú művet nem fog felismerni a hálózat a későbbiekben. Elmondható, hogy a vágás (*cutoff*) értékének növelésével nő a pontosság egy bizonyos mértékig, viszont egy ponton túl elkezdi csökkenni. Ez látható a 3.8 ábrán: az 1-es vágás értéknél látható, hogy a pontosság nagyon alacsony, tehát a hálózat nem tanul amennyiben csak a szófajok állnak számára rendelkezésre, szükségesek a szavak szóbeágyazásai is. Nagyságrendileg hasonló pontosságot érnek el azok az esetek, amikor $n \geq 100$. Érdemes észrevenni, hogy az az eset nem célravezető, amikor a teljes szótár összes szóbeágyazását felhasználjuk, ugyanis egy n érték felett elkezdi csökkenni a pontosság, emellett a statikus szóbeágyazások betöltése erőforrásigényes művelet, célszerű minimalizálni ezeknek a számát.



3.8. ábra. Hitschler és társainak eljárásának pontossága különböző vágási értékekkel

Meglepőnek tűnhet, hogy az eljárásnak elegendő a 100 leggyakoribb szó, hogy ennyire pontosan végezzen szerzőség-megállapítást. Ahogy a Lexikális stílusjegyek című fejezetben megemlítem, a funkció szavak használatának módja rendkívül jó jellemzőknek tekinthető. Az eljárás 100 vágási érték esetén emiatt eredményez nagyon pontos osztályozást az alacsony vágási érték ellenére. Az 3.1 táblázatban látható az n vágási érték esetén a halmazba eső legutolsó 20 elem gyakoriság szerint csökkenőbe rendezve.

$n = 100$	azonban, előtt, vagyok, alatt, király, ilyen, mindig, nincs, azért, lett, úr, amit, jól, szép, nekem, vele, éppen, e, mind, őket
$n = 5000$	bocsásson, szemembe, kendőt, tagjait, korábban, jöjjön, megérkeztek, elmaradt, kézre, láda, tót, állított, szépség, tartozó, szívből, szorgalmas, csúf, másképpen, értékes, sarkában
$n = 50000$	hegedűs, mindened, szemeid, gyerekeik, furulya, meghalna, kilátszó, unatkozom, kitölti, összkomfortos, ürességtől, nevezetesen, szobáról, munkaidőben, írtunk, telefonáló, inkognitó, címére, közlés, folyóiratokban
$n = N/A$	neveltetted, sajátom, telét, vészelttem, említettem-e, lelohasztják, regeneráláson, farod, felmehetnénk, regenerálódnunk, izgalmasat, odaszármaztak, küzdeniük, fejlődhettek, eltunyul, eltékozolhattuk, gyarapítjuk, életfeltételeket, cirádait, ökből

3.1. táblázat. n vágási érték esetén az elfogadott szavak halmazának utolsó 20 eleme

Ahogy az az 3.8. táblázatban látható az $n = 5000$ értékkel éri el a legnagyobb pontosságot az eljárás. Abban az esetben, ha csökkenteni szeretnénk a témafüggő jellemzők kinyerésének valószínűségét, érdemes csökkenteni az n értékét, ugyanis ezáltal a pontosság nem csökken jelentősen.

Szavak szótari vagy képzett alakjának használata

Hitschler és társainak eljárása a szavak képzett alakját használja fel. A magyar nyelv rugalmas szóképzési módjai miatt egy szó akár 20-30 különféle szuffixummal is előfordulhat a korpuszban. Emellett, a szótári alak használata a kétértelmű szavak során felmerülő hibák problémáját is részben kiküszöbölné. Ahhoz, hogy ezeket a szavakat azonosként kezeljük, a szavak képzett alakja helyett használhatnánk a szavak szótári alakját (*lemmáját*). A szótári alak kinyerésére külön eljárás nem szükséges, ugyanis a magyarlanc szövegelemző eszköz a kimeneti állomány egyik sorában az adott szó szótári alakját is feltünteti.

Hitschler és társainak eljárását kiértékeltem több különböző n vágási értékre és különböző hosszúságú szövegszekciókra. Az esetek mindegyikére megvizsgáltam

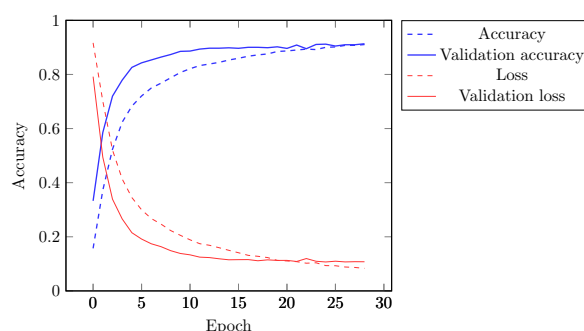
a szótári és a képzett alak használatát. A kiértékelés eredménye látható az 3.2 táblázatban, ahol az L a szótári alak, míg az O a képzett alak használatát jelzi. A táblázatban vastagon szedve emeltem ki a pontosabbnak bizonyult eredményt. A táblázatot megvizsgálva észrevehetjük, hogy a szótári alak használata jobban teljesít a képzett alakkal szemben, azonban a különbség a két variáció között a legtöbb esetben nem jelentős.

szövegrészlet hossza szavak alakja	150 szó (~5 mondat)		900 szó (~30 mondat)		3000 szó (~100 mondat)	
	O	L	O	L	O	L
pontosság, ha $n = 100$	70,88%	70,99%	91,72%	89,93%	88,89%	91,27%
pontosság, ha $n = 5000$	89,70%	91,19%	96,42%	95,08%	98,41%	94,44%
pontosság, ha $n = 50000$	91,63%	92,62%	95,97%	96,19%	86,51%	88,89%
pontosság, ha $n = N/A$	89,19%	90,43%	93,96%	95,30%	76,19%	94,44%

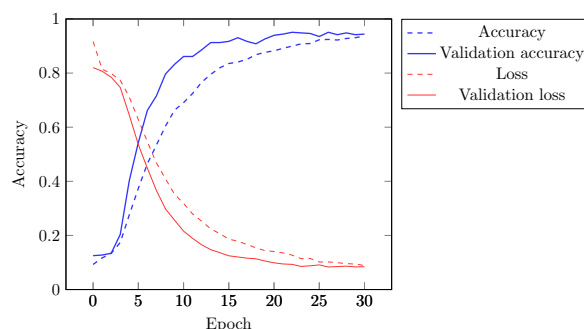
3.2. táblázat. Hitschler és társainak adaptált eljárásának pontossága különböző mondathosszak, vágási értékek és szóalakok esetén

Eljárás tanítása

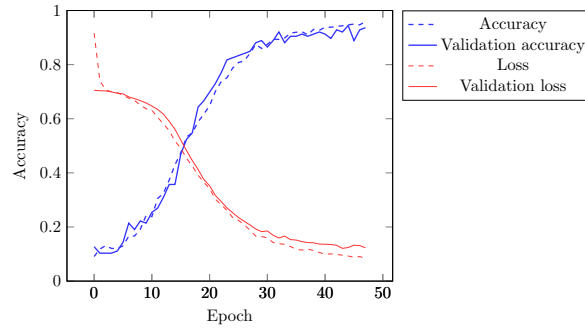
Ahogy az előző alfejezetekben ismertettem, a Hitschler és társainak általam adaptált eljárása a legjobban $n = 5\,000$ vágási érték és a szó szótári alakjának használatával bizonyult a legpontosabbnak, így az 3.9, 3.10, 3.11 ábrák az eljárás tanításának folyamatát ábrázolják ezen paraméterek mellett különböző szöveghosszak esetén.



3.9. ábra. Hitschler és társainak modelljének tanítása 5 mondat hosszúságú adatokon



3.10. ábra. Hitschler és társainak modelljének tanítása 30 mondat hosszúságú adatokon



3.11. ábra. Hitschler és társainak modelljének tanítása 100 mondat hosszúságú adatokon

A 3.9, 3.10, 3.11 ábrákat megvizsgálva megállapíthatjuk, hogy az eljárás bármely szöveghossz esetén képes pontosan megtanulni és megragadni a szerzői stílust. 5 és 30 mondat hosszúságú szövegszegmenseken az eljárás körülbelül 30 iteráción keresztül tanult, illetve megfigyelhető, hogy a validációs pontosság mindkét esetben már 10 iteráció után 80% felett van. 100 mondat hosszúságú szövegszegmensek esetén az iterációk száma 50 körül mozog. Emellett az látható a 3.11 ábrán, hogy a validációs pontosság lassabban növekszik, mint az első két esetben. Ennek ellenére mindhárom esetben az eljárás 90% feletti validációs pontosságot eredményez 15 osztályra (5-mondathosszúságú adatokon 91,19%, 30-mondathosszak esetén 95,08% és 100-mondathossz esetén 94,44%).

3.3.4. Kiértékelés eredményei

Az általam begyűjtött és annotált 15-szerzős magyar nyelvű korpuszon kiértékelt eljárások eredményei láthatók a 3.3 táblázatban. Az l értéke a szövegszegmensek hosszát jelöli, míg Z, J és H betűk az eljárások eredeti szerzőjének első kezdőbetűjét (Z - Zhang és társai, J - Jafariakinabad és társai, H - Hitschler és társai). A 3.3 táblázat második sorában az egy szerzőre jutó szövegszegmensek számát jelöli. Belátható, hogy a szövegszegmensek méretének növelésével csökkenni fog a szövegszegmensek száma, ami a táblázatból is kiolvasható.

Az iterációk számát megvizsgálva észrevehetjük, hogy a szövegszegmensek hosszától függetlenül Hitschler és társainak eljárása tanult a legtovább. Érdekes megjegyezni, hogy Zhang és társai illetve, Jafariakinabad és társai eljárásai közül az előbbi futott több iteráción keresztül rövidebb szövegszegmenseken, míg Jafariakinabad és társainak eljárása hosszabb szövegeken tanult tovább.

Az adatok pontosságát megfigyelve megállapítható, hogy a begyűjtött adathalmazon szövegszegmensek hosszától függetlenül Hitschler és társainak eljárása a legpontosabb, bármely mondat hossz esetén meghaladja a 90%-t. A másik két eljárás pontossága megközelítőleg sem eredményez hasonló teljesítményt.

	$l = 5$			$l = 30$			$l = 100$		
adat mérete szerzőnként	3900			653			199		
eljárás	Z	J	H	Z	J	H	Z	J	H
iterációk száma	14	10	27	12	13	31	9	18	48
pontosság	52,42%	35,95%	91,19%	35,37%	59,74%	95,08%	34,28%	59,31%	94,44%

3.3. táblázat. Magyar adatokon kiértékelt eljárások mutatói és eredményei

Érdemes megjegyezni, hogy Hitschler és társainak eljárása szófajcímkéző eszközt igényel, valamint előre tanított szóbeágyazásokat. Jafariakinabad és társainak eljárása implementációja során csak szófajcímkézőre van szükség, míg Zhang és társainak eljárásához nem szükséges külső eszköz használata. A magyarulanc vagy bármely másik szófajcímkéző eszköztár futtatása jelentősen megnöveli a tanítás és kiértékelés teljes futási idejét, ugyanis az adatok előkészítése egy igen hosszadalmas folyamat. Ugyan ez igaz Jafariakinabad és társainak eljárására. Emellett, a *Szeged word vectors* szóbeágyazási modell vektorai egy 4 GB méretű állomány formájában elérhetők. Egy kisebb teljesítményű számítógép, melynek memóriája 8 GB nem rendelkezik elegendő erőforrással a modell betöltésére, ami az eljárás használatának még egy korlátozó tényezője.

Az egyes eljárások tanítását különböző platformokon és számítógépeken végeztém, így az iterációk futási ideje nem hasonlítható össze követlenül. Ennek ellenére elmondható, hogy a konvolúciós neurális hálózatok nagyon jól párhuzamosíthatók, így videókártyán gyorsan végezhető a tanítás. Ennek következtében Zhang, valamint Hitschler és társainak eljárása rendkívül gyorsan tanult. Ezzel szemben Jafariakinabad és társainak eljárása LSTM és GRU hálózatokat alkalmaz, melyek a szekvenciák időbeli változásában keresnek összefüggést. A hálózat tehát az adatok egymásutániségát vizsgálja, így a műveletek a temporális függőségek miatt kevésbé párhuzamosíthatók. Ennek következménye, hogy az eljárás iterációinak futási ideje rendkívül nagy videókártya használatával is.

Összeségében belátható, hogy Hitschler és társainak eljárása biztosítja a legpontosabb eredményt, amennyiben rendelkezésünkre állnak a szükséges számítógépes erőforrások. A konvolúciós architektúrának köszönhetően a műveletek jól párhuzamosíthatók, így a modell nagyon gyorsan betanítható grafikus kártya használatával.

4. fejezet

Összefoglalás

Dolgozatomban a szerzőség-megállapítás lehetőségeit vizsgálom idegen nyelvű szövegekre, elemzem a megismert eljárások magyar nyelvű szövegekre való áttervezés, adaptálás lehetőségét. A kutatásom korábbi részeként lexikális és szintaktikai stílusjegyek használatát vizsgáltam meg magyar nyelvű adathalmazokon. Ennek következtében Zhang és Jafariakinabad és társainak eljárását adaptáltam magyar nyelvre. A kutatásom múltbeli mérföldköveinek egyike egy szerzőség-megállapításra alkalmas magyar nyelvű annotált adathalmaz begyűjtése volt. A manuálisan begyűjtött adathalmaz kicsinek bizonyult így a dolgozatom új eredményei közé sorolható egy olyan eljárás megtervezése és implementációja, mely egy szerző lista alapján letölti a Magyar Elektronikus Könyvtár HTML formátumban elérhető műveit, annotálja azokat, majd zajsűrűséget és adattisztítást végez.

Emellett, új kutatási eredménynek tekintem a lexikális és szintaktikai stílusjegyeiktől eltérő jellemzők megismerését, mint például a szóreprezentációkat. A szóbeágyazási modellek előnyeit, a Word2Vec eljárást, fastText kiterjesztést, transfer learning alapötletét ismertetem a dolgozatomban, valamint megvizsgálom milyen előretanított szóbeágyazások léteznek magyar nyelvre.

A kutatásom újabb mérföldköveinek legnagyobb eredménye Hitschler és társai eljárásának adaptációja magyar nyelvre. Az eredeti modell a statikus szóbeágyazások és szófajok együttesét ágyazza be, míg az általam implementált változatban a modell külön teszi ezt a két típusú adatra. Véleményem szerint részben ebben rejlik az eljárás sikeressége magyar nyelvű adatokon, ami eléri a 95%-os pontosságot 15 szerző esetén. Emellett megvizsgáltam az eljárás működését 5 különböző vágási érték mentén, ami a korpuszból felhasznált leggyakoribb szavak mennyiségét határozza meg. Ezen kívül megvizsgáltam, mely adattal végezhető pontosabban az elemzés: a szó szótári alakjával vagy a szó eredeti, képzett alakjával. Végezetül elvégzem az eljárások kiértékelését több hosszúságú adaton, valamint összehasonlítom az eredményekre vonatkozó metrikákat.

A kutatásomat az Emberi Erőforrások Minisztériuma támogatja, az ÚNKP-21-2 kódszámú, Új Nemzeti Kiválóság Program keretében.

4.1. További kutatási tervek

További kutatási terveim közé tartozik a szerzőség-megerősítésre alkalmazott eljárások vizsgálata, ahol akár az eredeti adathalmazon kívüli szerzőktől származó adatokról is megállapítható, hogy az adott szövegszegmensek szerzője azonos-e.

A szerzőség-megerősítés során a *metric learning* módszereit szeretném alkalmazni, melynek megismerése, vizsgálata az adott problémára szintén a további kutatási terveim közé tartozik.

Ábrajegyzék

2.1. Egy magyar nyelvű mondat konstituensfája	9
2.2. Egy magyar nyelvű mondat függőségi fája	10
2.3. A <i>reggeli</i> szó környezete a vektortérben 2-dimenzióra projektálva [17]	12
2.4. Skip-gram modell felépítése [20]	13
2.5. Szövegelemző eljárások fejlődése	16
2.6. Egyszerű konvolúciós architektúra [36]	18
2.7. Karakterszintű konvolúciós hálózatok fejlődése	19
2.8. Zhang és társai által javasolt konvolúciós modell felépítése [31]	20
2.9. Hitschler és társai által javasolt konvolúciós modell felépítése [34]	23
2.10. Jafariakinabad és társai által javasolt modell felépítése [33]	24
2.11. Chen és társainak által javasolt mondatszintű modell felépítése [35]	25
2.12. Chen és társai által javasolt cikkszintű modell felépítése [35]	25
3.1. Zhang és társainak modelljének tanítása 5 mondat hosszúságú adatokon	33
3.2. Zhang és társainak modelljének tanítása 30 mondat hosszúságú adatokon	33
3.3. Zhang és társainak modelljének tanítása 100 mondat hosszúságú adatokon	34
3.4. Jafariakinabad és társainak modelljének tanítása 5 mondat hosszúságú adatokon	35
3.5. Jafariakinabad és társainak modelljének tanítása 30 mondat hosszúságú adatokon	35
3.6. Jafariakinabad és társainak modelljének tanítása 100 mondat hosszúságú adatokon	35
3.7. Hitschler és társainak általam adaptált eljárása	36
3.8. Hitschler és társainak eljárásának pontossága különböző vágási értékekkel	38
3.9. Hitschler és társainak modelljének tanítása 5 mondat hosszúságú adatokon	39
3.10. Hitschler és társainak modelljének tanítása 30 mondat hosszúságú adatokon	39
3.11. Hitschler és társainak modelljének tanítása 100 mondat hosszúságú adatokon	40

Táblázatok jegyzéke

2.1.	magyarlanc <i>depparse</i> módú elemzésének kimenete	15
2.2.	Zhang és társainak modelljének rétegeinek paraméterei	20
2.3.	Megvizsgált eljárások összefoglaló táblázata	27
3.1.	n vágási érték esetén az elfogadott szavak halmazának utolsó 20 eleme	38
3.2.	Hitschler és társainak adaptált eljárásának pontossága különböző mon- dathosszak, vágási értékek és szóalakok esetén	39
3.3.	Magyar adatokon kiértékelt eljárások mutatói és eredményei	40

Irodalomjegyzék

- [1] Dobi Jan Sándor, Mészáros Tamás és Kiss Margit. “Shtylo: stilometriai elemzések webes támogatása”. ().
- [2] Robert AJ Matthews és Thomas VN Merriam. “Neural computation in stylo-metry I: An application to the works of Shakespeare and Fletcher”. *Literary and Linguistic computing* 8.4 (1993), 203–209. old.
- [3] Petr Plecháč. “Relative contributions of Shakespeare and Fletcher in Henry VIII: An analysis based on most frequent words and most frequent rhythmic patterns”. *Digital Scholarship in the Humanities* 36.2 (2021), 430–438. old.
- [4] Kiss Margit. “Stilometriai elemzés lehetőségei magyar történeti szövegtörzsön”. *DIGITÁLIS BÖLCSESZET* 2 (2019), 15–33. old.
- [5] Carol Rounds. *Hungarian: An essential grammar*. Routledge, 2009.
- [6] John Lyons. *Natural Language and Universal Grammar: Essays in Linguistic Theory*. 1. köt. Cambridge University Press, 1991. DOI: 10.1017/CB09781139165877.
- [7] Gobinda G. Chowdhury. “Natural language processing”. *Annual Review of Information Science and Technology* 37.1 (2003), 51–89. old. DOI: <https://doi.org/10.1002/aris.1440370103>. eprint: <https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/aris.1440370103>. URL: <https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440370103>.
- [8] Rong Zheng, Jiexun Li, Hsinchun Chen és Zan Huang. “A framework for authorship identification of online messages: Writing-style features and classification techniques”. *Journal of the American Society for Information Science and Technology* 57.3 (2006), 378–393. old. DOI: <https://doi.org/10.1002/asi.20316>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20316>.
- [9] Sara El Manar El és Ismail Kassou. “Authorship analysis studies: A survey”. *International Journal of Computer Applications* 86.12 (2014).
- [10] Maciej Eder. “Style-markers in authorship attribution: a cross-language study of the authorial fingerprint”. *Studies in Polish Linguistics* 6.1 (2011).
- [11] Monika Szirmai. *Bevezetés a korpusznelvészetbe*. Tinta könyvkiadó, 2005.
- [12] Efstathios Stamatatos. “A survey of modern authorship attribution methods”. *Journal of the American Society for information Science and Technology* 60.3 (2009), 538–556. old.
- [13] Fóris Ágota. “Lexikológiai és lexikográfiai ismeretek magyar (mint idegen nyelv) tanároknak”. *Károli könyvek. Magyar nyelv-tanári segédkönyvek.* Károli Gáspár Református Egyetem–L’Harmattan Kiadó, Budapest (2018).

- [14] Zoltán Subecz. “Események detektálása, osztályozása és szemantikus szerepeik címkézése”. Dissz. szte, 2020.
- [15] Zsibrita János, Vincze Veronika és Farkas Richárd. “magyarlanc: A toolkit for morphological and dependency parsing of Hungarian”. *Proceedings of RANLP 2013* (2013), 763–771. old.
- [16] Mohammed A El Affendi és KHS Al Rajhi. “Text encoding for deep learning neural networks: A reversible base 64 (Tetrasexagesimal) Integer Transformation (RIT64) alternative to one hot encoding with applications to Arabic morphology”. *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*. IEEE. 2018, 70–74. old.
- [17] *Magyar Tudomány 2018/7*. Akadémiai Kiadó, 2018. DOI: 10 . 1556/. URL: https://mersz.hu/hivatkozas/matud_f8922_p1#matud_f8922_p1.
- [18] Tomas Mikolov, Kai Chen, Greg Corrado és Jeffrey Dean. “Efficient estimation of word representations in vector space”. *arXiv preprint arXiv:1301.3781* (2013).
- [19] Yoav Goldberg és Omer Levy. “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method”. *arXiv preprint arXiv:1402.3722* (2014).
- [20] Jie Hu, Shaobo Li, Yong Yao, Liya Yu, Yang Guanci és Jianjun Hu. “Patent Keyword Extraction Algorithm Based on Distributed Representation for Patent Classification”. *Entropy* 20 (2018. febr.), 104. old. DOI: 10 . 3390 / e20020104.
- [21] Piotr Bojanowski, Edouard Grave, Armand Joulin és Tomas Mikolov. “Enriching word vectors with subword information”. *Transactions of the association for computational linguistics* 5 (2017), 135–146. old.
- [22] Armand Joulin, Edouard Grave, Piotr Bojanowski és Tomas Mikolov. “Bag of Tricks for Efficient Text Classification”. *arXiv preprint arXiv:1607.01759* (2016).
- [23] Karl Weiss, Taghi M Khoshgoftaar és DingDing Wang. “A survey of transfer learning”. *Journal of Big data* 3.1 (2016), 1–40. old.
- [24] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta és Thomas Wolf. “Transfer Learning in Natural Language Processing”. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. jún., 15–18. old. DOI: 10 . 18653 / v1 / N19 – 5004. URL: <https://aclanthology.org/N19-5004>.
- [25] György Orosz. *Hungarian NLP tools*. Utóljára megtekintve: 2022. URL: <https://github.com/oroszgy/awesome-hungarian-nlp>.
- [26] Veronika Vincze, Katalin Ilona Simkó, Zolt Szántó és Richárd Farkas. “Universal Dependencies and Morphology for Hungarian-and on the Price of Universality”. Association for Computational Linguistics. 2017.
- [27] Marie-Catherine de Marneffe és tsai. “Universal Dependencies: A cross-linguistic typology”. ().

- [28] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber és Laura E Barnes. “Hdltex: Hierarchical deep learning for text classification”. *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE. 2017, 364–371. old.
- [29] Li Deng és Yang Liu. “A joint introduction to natural language processing and to deep learning”. *Deep learning in natural language processing*. Springer, 2018, 1–22. old.
- [30] Yoon Kim. *Convolutional Neural Networks for Sentence Classification*. 2014. arXiv: 1408.5882 [cs.CL].
- [31] Xiang Zhang, Junbo Zhao és Yann LeCun. “Character-level convolutional networks for text classification”. *Advances in neural information processing systems* 28 (2015), 649–657. old.
- [32] Sebastian Ruder, Parsa Ghaffari és John G Breslin. “Character-level and multi-channel convolutional neural networks for large-scale authorship attribution”. *arXiv preprint arXiv:1609.06686* (2016).
- [33] Fereshteh Jafariakinabad, Sansiri Tarnpradab és Kien A Hua. “Syntactic recurrent neural network for authorship attribution”. *arXiv preprint arXiv:1902.09723* (2019).
- [34] Julian Hitschler, Esther Van Den Berg és Ines Rehbein. “Authorship attribution with convolutional neural networks and POS-eliding”. *Proceedings of the Workshop on Stylistic Variation*. 2017, 53–58. old.
- [35] Chen Qian, Ting He és Rao Zhang. “Deep Learning based Authorship Identification”. 2017.
- [36] Keiron O’Shea és Ryan Nash. “An Introduction to Convolutional Neural Networks”. *CoRR* abs/1511.08458 (2015). arXiv: 1511.08458. URL: <http://arxiv.org/abs/1511.08458>.
- [37] Apeksha Shewalkar, Deepika Nyavanandi és Simone A Ludwig. “Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU”. *Journal of Artificial Intelligence and Soft Computing Research* 9.4 (2019), 235–245. old.
- [38] *Magyar Elektronikus Könyvtár*. Utóljára megnyitva: 2022. URL: <https://mek.oszk.hu/>.
- [39] Wikipedia. *Magyar költők, írók listája*. Utóljára megtekintve: 2022. URL: https://hu.wikipedia.org/wiki/Magyar_k%C3%B6lt%C5%91k,_%C3%ADr%C3%B3k_list%C3%A1ja.
- [40] <https://github.com/BrambleXu/nlp-beginner-guide-keras/tree/master/char-level-cnn>. Utóljára megtekintve: 2022.
- [41] <https://github.com/danish241194/Syntactic-Recurrent-Neural-Network-for-Authorship-Attribution>. Utóljára megtekintve: 2022.
- [42] Liang Xu. *Keras Implementations for NLP Models*. Utóljára megtekintve: 2022. 2018. URL: <https://github.com/BrambleXu/nlp-beginner-guide-keras>.