FAKULTETA ZA INFORMACIJSKE ŠTUDIJE
V NOVEM MESTU

D I P L O M S K A N A L O G A

VISOKOŠOLSKEGA STROKOVNEGA ŠTUDIJSKEGA PROGRAMA
PRVE STOPNJE

ANDRAŽ PELICON

FAKULTETA ZA INFORMACIJSKE ŠTUDIJE
V NOVEM MESTU

DIPLOMSKA NALOGA

ZAZNAVANJE SENTIMENTA V NOVICAH Z GLOBOKIMI NEVRONSKIMI MREŽAMI

Mentorica: doc. dr. Petra Kralj Novak
Somentorica: izr. prof. dr. Biljana Mileva Boshkoska

Novo mesto, september 2019 Andraž Pelicon

IZJAVA O AVTORSTVU

Podpisani Andraž Pelicon, študent FIŠ Novo mesto, izjavljam:

- da sem diplomsko nalogo pripravljal samostojno na podlagi virov, ki so navedeni v diplomski nalogi,
- da dovoljujem objavo diplomske naloge v polnem tekstu, v prostem dostopu, na spletni strani FIŠ oz. v elektronski knjižnici FIŠ,
- da je diplomska naloga, ki sem jo oddal v elektronski obliki identična tiskani verziji,
- da je diplomska naloga lektorirana.

V Novem mestu, dne ___22. 9. 2019_____ Podpis avtorja _____

Andraž Pelicon

Digitalno podpisal Andraž Pelicon
Datum: 2019.09.22 18:53:23 +02'00'

Zahvaljujem se mentoricama doc. dr. Petri Kralj Novak in izr. prof. Biljani Milevi Boshkoski za vso pomoč in nasvete pri izdelavi diplomske naloge.

Zahvaljujem se Mateju Martincu za vse ideje na področju, kjer ima veliko več izkušenj kot jaz.

Zahvaljujem se svoji družini, ki mi je vsa dolga leta študija stala ob strani.

Zahvaljujem se Evi, ker me je pregnala za pisalno mizo.

POVZETEK

Diplomska naloga se ukvarja z analizo sentimenta v novicah. To področje v zadnjem času pridobiva na priljubljenosti, predvsem v okviru napovedovanja gibanja finančnih trgov, vendar je za slovenski jezik še dokaj slabo raziskano. Za slovenščino sicer obstajajo modeli, osnovani na metodi podpornih vektorjev, vendar ti niso dostopni za javno uporabo.

V okviru te raziskave smo zasnovali arhitekturo na osnovi nevronskih mrež, ki za klasifikacijo uporablja kombinacijo samodejno generiranih značilk in TF-IDF obtežitev. Modeli, ki uporabljajo omenjeno arhitekturo, dosegajo primerljive rezultate z že obstoječimi modeli in so sposobni učinkovitega učenja na korpusih v velikosti okrog 10.000 dokumentov. Najuspešnejši model iz raziskave je na voljo kot spletna storitev na naslovu classify.ijs.si.

KLJUČNE BESEDE: analiza sentimenta, novice, slovenščina, nevronske mreže, globoko učenje

ABSTRACT

The present thesis deals with the sentiment analysis of news. This field has recently gained in popularity, especially as a supporting method for stock market prediction, but not much research has yet been done on the news in the Slovenian language. Models based on support vector machines do exist but are not available for public use.

We developed a neural network architecture that leverages both automatically generated features and TF-IDF weights for classification of Slovenian news. Models based on this architecture achieve comparable results with existing models and can be successfully trained on datasets of approximately 10,000 documents. Our best performing model is available for public use in the form of a web service on the URL classify.ijs.si.

KEYWORDS: sentiment analysis, news, Slovene, neural networks, deep learning

KAZALO

KAZALO SLIK IN TABEL

1 UVOD

Ljudje smo pogosto soočeni z odločitvami, kjer se zdi, da se samo na podlagi svojih informacij ne moremo ustrezno odločiti za eno od alternativ. V takih primerih se pogosto opremo na ljudi, ki jim zaupamo. Vzemimo na primer večje finančne investicije, kot sta nakup avtomobila ali stanovanja. Če se sami nikakor ne moremo odločiti med dvema modeloma avtomobila, pogosto vprašamo znanca, ki morda ima podoben model ali vsaj znamko avtomobila, kaj si o določenem nakupu misli, ali pa se za nasvet obrnemo na ocene strokovnjakov. Ko kupujemo stanovanje v novem kraju se morda obrnemo na prijatelje, ki tam že živijo, in jih povprašamo, ali je okolica primerna za vzgojo otrok in ali je dovolj zelenih površin za sprehajanje psa.

Mnenja in subjektivni odnos ali sentiment do določenega subjekta so ljudem zelo pomembni pri vsakodnevnem odločanju in sodobne tehnologije omogočajo hitro in učinkovito izmenjavo mnenj kjerkoli po svetu. Na družbenih platformah kot sta Twitter in Facebook ljudje neprestano izražajo mnenja o različnih izdelkih ter svoj odnos do raznih tem in oseb, od politikov do podjetnikov. Spletne trgovine omogočajo, da kupci o določenem izdelku oddajo svoje mnenje in oceno, kar pomaga tako drugim potencialnim kupcem kot trgovini sami, da lahko lažje oceni priljubljenost prodajanih izdelkov. Novičarske platforme omogočajo objavo komentarjev, kjer bralci izražajo svoja mnenja glede prebranih novic in subjektivne poglede na obravnavane teme. Prek vseh teh kanalov se tako proizvedejo ogromne količine prosto dostopnih besedil, ki jih je praktično nemogoče ročno pregledati. Zato se je že precej zgodaj pojavilo zanimanje za izdelavo modelov in sistemov, ki bi s pomočjo tehnik strojnega učenja bili sposobni samodejno analizirati mnenja v krajših in daljših oblikah besedil.

Najbolj zgodnji modeli, ki so se ukvarjali z analizo mnenj, so se osredotočali predvsem na samodejno prepoznavanje mnenj o posameznih izdelkih v ocenah in komentarjih uporabnikov, kjer je mnenje uporabnika dokaj jasno izraženo v besedilu in kjer mnenje poleg besedila izraža tudi številčna ali znakovna ocena. Z eksplozijo družbenih omrežij je za posamezna podjetja postalo zanimivo sledenje priljubljenosti svoje znamke v javnih objavah uporabnikov na spletu. Kmalu pa se je zanimanje začelo usmerjati tudi na vrste besedil, ki naj bi načeloma nevtralno podajale informacije in ne bi izražale nobenega mnenja avtorja ali njegovega subjektivnega

pogleda na obravnavano tematiko, kot so na primer novice. Analiza sentimenta v novicah v zadnjem času postaja vse bolj priljubljena na finančni domeni, kjer številni raziskovalci poskušajo na podlagi analiziranega sentimenta napovedati gibanja na finančnih trgih.

Večinoma se analiza mnenj in sentimenta zastavi kot klasifikacijski problem, kjer se posamezna besedila klasificira v eno od predhodno definiranih kategorij. Najbolj razširjene tehnike strojnega učenja na tem področju so metode podpornih vektorjev, vendar se v zadnjem času tudi na tem področju pojavlja vse več modelov globokih nevronskih mrež, ki dosegajo primerljive rezultate.

V zvezi s količino podatkov, ki je potrebna za učenje sposobnega modela, ni enotnega mnenja, saj je slednje precej odvisno od samega problema in velikosti modela. Kljub temu empirični rezultati iz posameznih študij kažejo, da se lahko na podlagi večje količine podatkov modeli strojnega učenja, predvsem modeli globokega učenja, naučijo bolj informativnih značilk, ki pomagajo pri boljši klasifikaciji. Yang, Yang, Dyer, He, Smola in Hovy (2016) so razvili arhitekturo globokih nevronskih mrež za klasifikacijo daljših dokumentov. Poročali so o izboljšanju rezultatov na vseh šestih preskusnih zbirkah podatkov, pri čemer sta dve manjši zbirki podatkov vsebovali več kot 300.000 primerov, tri večje podatkovne zbirke so vsebovale več kot milijon primerov, največja pa več kot tri milijone primerov. Devlin, Chang, Lee in Toutanova (2018) so na podlagi variante nevronskih mrež, tako imenovane pozornosti (ang. *attention*), razvili velik jezikovni model, imenovan BERT. V članku poročajo, da je bil model naučen na dveh sicer neoznačenih korpusih, BookCorpus, ki vsebuje 11.038 knjig v celoti in 800 milijonov besed, in korpusu člankov angleške Wikipedije, ki vsebuje 2.500 milijonov besed. Pridobivanje tako velikih količin podatkov, predvsem označenih, je pogosto zelo drag in zamuden postopek, zato je učenje uspešnih klasifikacijskih modelov na omejenih virih aktualen problem.

1.1 Cilji in hipoteze

Čeprav se analiza sentimenta v novicah vse pogosteje izvaja za večje jezike, je to področje za slovenščino zaenkrat še precej neraziskano. Edini modeli, ki po naših informacijah obstajajo, so modeli, naučeni na podlagi metode podpornih vektorjev. Cilj te diplomske naloge je, da bi razvili model globokih nevronskih mrež, ki bi uspešno napovedoval sentiment v novicah, kot ga dojema povprečen slovenski bralec. Pri tem si kot prvo hipotezo postavljamo trditev, da je

mogoče z modeli nevronskih mrež doseči primerljivo klasifikacijsko točnost kot z drugimi tehnikami strojnega učenja.

Za naš problem trenutno še ni na voljo velike količine kakovostno označenih podatkov. Tako bomo poskušali naše modele naučiti na korpusu, ki vsebuje okrog 10.000 podatkov. Naša druga hipoteza je, da je tudi s tako omejeno količino podatkov mogoče izdelati model globokih nevronskih mrež, ki uspešno modelira dani problem.

2 PREGLED PODROČJA

V tem poglavju bomo predstavili področje analize sentimenta. Najprej bomo v poglavju 2.1 podali splošen opis problema, v poglavju 2.2 pa bomo podali pregled raziskav, ki se izvajajo na tem področju.

2.1 Opis področja

Preden opredelimo, kaj zajema področje analize sentimenta, moramo najprej odgovoriti na vprašanje, kaj sploh je sentiment. Sentiment v besedilih so najprej opredelili jezikoslovci, in sicer kot vsak element, ki ga ni mogoče objektivno opazovati ali preveriti (Quirk, Greenbaum, Leech & Svartvik, 1985). Sem spadajo med drugim čustva, mnenja in predvidevanja. Sama opredelitev sentimenta nakazuje na možne težave pri njegovi analizi, saj subjektivni elementi v besedilih velikokrat niso neposredno izraženi, so zelo odvisni od konteksta, način njihovega izražanja pa se med posameznimi avtorji razlikuje (Mejova, 2009).

Analiza sentimenta, v literaturi imenovana tudi analiza mnenj, je večdisciplinarno področje na preseku računalništva in jezikoslovja, ki spada v širše področje obdelave naravnega jezika (*NLP*). Ukvarja se z analizo in prepoznavanjem odnosa, čustev in mnenj ljudi do določenega objekta. Zajema skupek tehnik iz računalniške obdelave naravnega jezika, s katerimi lahko iz besedila izvlečemo subjektivne informacije (Beigi, Maciejewski & Liu, 2015).

2 PREGLED PODROČJA

Tradicionalno se sentiment v besedilih razpoznava v odnosu do objekta sentimenta, tj. predmeta, pojma ali osebe, na katero se izraženi sentiment nanaša (Meyova, 2009). Primeri objektov sentimenta v tvitih so izdelki določenega podjetja, v ocenah filmov pa filmi. Liu (2012) omenja tri ravni, na katerih se sentiment običajno razpoznava:

- Raven dokumenta, kjer je naloga razpoznati, ali celoten dokument izraža negativen ali pozitiven sentiment do objekta. V oceni izdelka želimo na primer razpoznati, kakšno mnenje ima avtor ocene o izdelku.
- Raven povedi, kjer je cilj določiti sentiment, ki ga do objekta izražajo posamezne povedi. Povedi, ki do objekta ne izražajo nobenega mnenja, so označene kot nevtralne.
- Raven entitete in vidika, na kateri se izvaja še podrobnejša analiza sentimenta. Glavna predpostavka namreč je, da z analizo na celotnem dokumentu ali povedi ne moremo vedno natančno določiti, kakšen odnos ima analizirana struktura do objekta. Poved "Čeprav postrežba ni na nivoju, mi je restavracija kljub temu všeč", sicer ima pozitiven podton, ne moremo je pa označiti kot povsem pozitivno, saj ima pozitivno mnenje do restavracije in negativno do postrežbe v dani restavraciji. Analiza sentimenta na tej ravni si tako prizadeva prepoznati različne objekte sentimenta, ki nastopajo v neki jezikovni strukturi, ter prepoznati mnenja, ki jih do teh objektov izraža posamezna jezikovna struktura.

Čeprav je tako razmerje med sentimentom v jezikovni strukturi in objektom sentimenta precej razširjeno, pa se nekateri avtorji problema lotijo z druge perspektive. Namesto vprašanja, kakšno mnenje ima avtor do nekega objekta v besedilu, se sprašujejo, kakšen sentiment zaznavajo bralci v določenem besedilu. Primer takega razumevanja vloge sentimenta najdemo v Lin, Yang in Chen (2008). Kot bomo videli pri opisu podatkov v poglavju 4.1, se naša raziskava osredotoča na slednjo perspektivo sentimenta, in sicer na nivoju dokumenta.

S prehodom v novo tisočletje se je področje analize sentimenta izjemno razširilo, k čemur je pripomogla tudi ogromna zbirka subjektivnih vrst besedil, ki jih je mogoče pridobiti prek interneta (Liu, 2015). Tako se je analiza sentimenta postopoma razširila na različne vrste besedil in problemov, tako v raziskovalnih vodah kot v aplikacijah za komercialne namene. Hu in Liu (2004) sta tehnike analize sentimenta uporabljala za pridobivanje mnenja o izdelkih iz ocen kupcev, Tripathy, Agrawal in Rath (2016) pa za rudarjenje mnenj o filmih iz ocen gledalcev na spletni platformi IMDB. Poleg analize mnenj o izdelkih se analiza sentimenta uporablja tudi za prepoznavanje mnenj v političnih debatah. V času predsedniških volitev v Sloveniji leta 2012

je podjetje Gama Systems razvilo platformo, ki je v realnem času analizirala objave na družbenem omrežju Twitter ter v njih analizirala mnenja in odzive gledalcev na soočenja med predsedniškimi kandidati. Za vsakega kandidata je nato prikazala politični indeks, ki je bil opredeljen kot absolutna razlika med številom pozitivnih in negativnih mnenj v objavah. Platforma je bila uporabljena med predsedniškimi soočenji na televizijskem kanalu Pop TV (24ur.com, 2012).

V vseh navedenih primerih se je analiza sentimenta uporabljala na tipih besedil, v katerih je sentiment bolj ali manj eksplicitno izražen. Nasprotno se v novicah avtorji trudijo dati vtis objektivnosti, zato se pogosto izogibajo uporabi povsem negativnih ali pozitivnih izrazov. Svoja mnenja, če jih že izražajo, poskusijo posredovati na bolj impliciten način, na primer tako, da nekatera dejstva bolj poudarijo, navajajo citate istomislečih oseb ali mnenja izrazijo s kompleksnejšimi argumentativnimi strukturami (Balahur in drugi, 2013). Iz teh razlogov je analiza mnenj v novicah in sorodnih vrstah besedil precej težja. Številni avtorji so se lotili tudi tega problema, predvsem na področju napovedi gibanj finančnih trgov. De Kauter, Breesch in Hoste (2015) navajajo, da imajo novice neposreden vpliv na finančne trge, saj dobre novice načeloma spodbudijo, slabe novice pa omejijo rast na trgu. V istem članku navajajo tudi, da imajo negativne novice načeloma večji vpliv na rast kot dobre novice. Tako je postala analiza sentimenta v novicah priljubljena na finančnem področju, kjer se analizira sentiment v novicah, ki se dotikajo makroekonomskih ali političnih tem oziroma vsebujejo podatke o posameznih podjetjih. Vse te informacije so namreč lahko relevantne za opis trga (Cortis in drugi, 2017).

Tehnike analize sentimenta navadno ločimo na dve kategoriji glede na pristop. Leksikalni pristopi se opirajo na leksikone sentimenta, ki vsebujejo besede in besedne zveze, katerim je pripisan sentiment, ki ga izražajo. Določenemu besedilu lahko tako s pomočjo leksikona določimo sentiment na podlagi besed in besednih zvez, ki jih vsebuje. Primer leksikona sentimenta za angleški jezik je SentiWordNet, (Baccianella, Esuli & Sebastiani, 2010) kjer je vsak vnos sorazmerno označen s tremi ravnmi sentimenta, negativno, pozitivno ali nevtralno, glede na to, kakšen sentiment vnos izraža. Zanimiv leksikon sentimenta je Emoji Sentiment Ranking, (Kralj Novak, Smailović, Sluban & Mozetič, 2015) ki vsebuje oceno sentimenta za 751 najpogosteje uporabljenih emojijev na platformi Twitter. Sestavljen je na podlagi objav na Twitterju v 13 evropskih jezikih in se tako lahko uporablja kot jezikovno neodvisen leksikon sentimenta za evropske jezike.

Liu (2015) navaja, da imajo pristopi, ki se opirajo izključno na leksikone sentimenta, precej pomanjkljivosti. Besedam in besednim zvezam se lahko sentiment spreminja glede na tematiko

in področje besedila. Poleg tega lahko besedila, ki vsebujejo precej besed in besednih zvez z močno izraženim sentimentom, sama po sebi ne izražajo nobenega sentimenta ali mnenja. Primer za to so običajno vprašanja, kot na primer “Ali mi lahko kdo predlaga dober fotoaparat?”. Poved sicer vsebuje besedo “dober”, ki sama po sebi izraža pozitiven sentiment, vendar je poved v celoti nevtralna. Dodatne težave se pojavijo s prepoznavanjem sarkazma. Take povedi lahko vsebujejo besede določenega sentimenta, izražajo pa povsem nasproten sentiment. Kot primer vzemimo poved “Res super avto, po dveh dneh je nehal delati!” Končno imamo lahko tudi povedi, ki sicer ne vsebujejo besed ali besednih zvez z jasno izraženim sentimentom, ki pa o cilju vseeno posredujejo neko mnenje. Na primer poved “Ta pralni stroj porabi veliko vode,” kljub uporabi nevtralnih izrazov jasno izraža negativno mnenje o pralnem stroju.

Drugi pristop, ki se vse pogosteje uporablja, je pristop z uporabo tehnik strojnega učenja. Pri teh pristopih se uporablja korpus besedil, v katerem je običajno vsako besedilo označeno s sentimentom, ki ga izraža, za učenje klasifikatorja. Naučeni klasifikator se nato uporabi za samodejno analizo sentimenta v novih besedilih.

2.2 Pregled obstoječih raziskav

Na področju analize sentimenta v novicah sta uporabljena oba pristopa, opisana v prejšnjem poglavju. Taj, Shaikh in Meghji (2019) so sestavili svoj korpus iz novic, objavljenih na portalu BBC, ter s pomočjo TF-IDF obtežitev v vsaki novici poiskali ključne besede. Nato so izključno na podlagi podatkov o sentimentu ključnih besed iz leksikona sentimenta poskušali analizirati sentiment v posameznih novicah. Bolj priljubljen pristop na tem področju je uporaba tehnik strojnega učenja, v okviru katerih se za učenje modelov najpogosteje uporablja metoda podpornih vektorjev (ang. *support vector machine*). Kaur in Kaur (2017) sta s pomočjo metode podpornih vektorjev analizirala sentiment v novicah, napisanih v jeziku pandžabi, pri čemer sta besedila obdelala s standardnimi metodami predobdelave, in sicer tokenizacijo, odstranjevanjem neinformativnih besed in krnjenjem. Pogosto se v kombinaciji s strojnim učenjem uporabljajo tudi leksikoni sentimenta za podajanje dodatnih informacij o sentimentu v model. Lin, Yang in Chen (2009) so analizirali sentiment v kitajskih novicah, kjer so uporabili metodo podpornih vektorjev, pri čemer so kot vhod uporabili več vrst značilk, ki so jih pridobili s kombinacijo predobdelave besedil v korpusu in informacij iz leksikona sentimenta. Kot značilke so uporabili bigrame pismenk, posamezne besede, metapodatke o posamezni novici,

pripone besed in sentiment posameznih besed, ki so ga pridobili v leksikonu sentimenta. Li, Xie, Chen, Wnag in Deng (2014) so podobno metodo uporabili za preverjanje vpliva novic na donosnost delnic. Naloga modelov je bila predvideti gibanje cen delnic, pri čemer je bila učna množica sestavljena iz novic, kot napovedni cilj pa so uporabili historične cene delnic iz istega petletnega obdobja, v katerem so izšle novice iz učnega korpusa. Da bi v model vnesli podatke o sentimentu, so s pomočjo leksikona sentimenta sestavili matriko sentimenta za celotno besedišče. Nato so matriko sentimenta uporabili za preslikavo TF-IDF matrike, ki so jo sestavili na podlagi pogostosti izrazov v korpusu novic, v nov vektorski prostor. Dobljeno matriko so nato uporabili za učenje modela s pomočjo metode podpornih vektorjev. Kumar, Sethi, Akhtar, Ekbal, Biemann in Bhattacharyya (2017) so novicam poskušali določiti sentiment kot zvezno vrednost v intervalu med -1 in 1, pri čemer je vrednost -1 predstavljala najbolj negativen sentiment, vrednost 1 pa najbolj pozitiven sentiment. Tudi oni so za učenje uporabili metodo podpornih vektorjev s to razliko, da so problem opredelili kot regresijski problem. Za napoved sentimenta so kot značilke uporabili uni- in bigrame besed, obtežene s TF-IDF utežmi, značilke, izdelane na podlagi informacij iz leksikona sentimenta, in 300-dimenzijske vektorske vložitve besed word2vec.

V zadnjem času se tudi na področju analize sentimenta vedno bolj pojavljajo modeli globokega učenja, ki temeljijo na nevronskih mrežah. Mansar, Gatti, Ferradanas, Guerini in Staiano (2017) so za analizo sentimenta uporabili konvolucijske nevronske mreže, ki se sicer pogosteje uporabljajo v robotskem vidu. S pomočjo konvolucijskih plasti so pridobili notranje predstavitve posamezne novice iz učnega korpusa in jih nato združili z oceno sentimenta posamezne novice, ki so jo pridobili s preprostim modelom VADER, ki temelji na pravilih. Tako pridobljene značilke so nato uporabili kot vhod v končni klasifikator, ki je temeljil na polnopovezani nevronski mreži. Razen tokenizacije se metod predobdelave besedil niso posluževali. Njihov model je bil leta 2017 najboljši v eni od nalog na letnem mednarodnem tekmovanju o računalniški semantični analizi SemEval. Moore in Rayson (2017) sta za namene analize sentimenta iz naslovov novic izdelala dva modela, enega na podlagi metode podpornih vektorjev z ročno generiranimi značilkami, drugega pa na podlagi dvosmernih rekurentnih nevronskih mrež s dolgo-kratkoročno pomnilno celico (ang. *bidirectional long-short term*

memory ali *BILSTM*). V svojem eksperimentu poročata, da je njihov nevronski model dosegel za 4–6 odstotkov višjo točnost v primerjavi z metodo podpornih vektorjev.

Področje analize sentimenta v novicah v slovenskem jeziku je zaenkrat še precej neraziskano. Bučar, Povh in Žnidaršič (2016) so na korpusu novic, ki so ga sami sestavili in tudi javno objavili (Bučar, Povh & Žnidaršič, 2018), naučili več modelov na podlagi metod naivnega Bayesa, podpornih vektorjev in k najbližjih sosedov, pri čemer so kot značilke uporabljali n-grame besed s TF-IDF obtežitvami. Najbolj učinkovite modele, ki so bili naučeni za razlikovanje sentimenta glede na tri kategorije, negativno, nevtralno in pozitivno, smo v tej diplomski uporabili kot osnovo za primerjavo naših naučenih modelov.

3 OSNOVE ANALIZE SENTIMENTA BESEDIL

V tem poglavju predstavimo tehnike in algoritme, ki se uporabljajo pri računalniški analizi sentimenta. Najprej predstavimo različne postopke predobdelave besedil, ki pomembno vplivajo na končno uspešnost klasifikacijskih modelov. Nato predstavimo metode vektorizacije besedil, s katerimi besedila spremenimo v obliko, ki jo zahtevajo klasifikacijski algoritmi. Nadaljujemo s predstavitvijo načinov vnašanja dodatnih informacij o sentimentu besedil v model, ki naj bi modelu pomagale pri lažji klasifikaciji. Poglavje zaključimo z opisom izbranega klasifikacijskega algoritma, tj. nevronskih mrež.

3.1 Predobdelava besedil

Številni problemi na področju analize sentimenta so zastavljeni kot klasifikacijski problemi (Meyova, 2009). Učenje klasifikacijskih modelov za te probleme je navadno nadzorovano, kar pomeni, da se modeli učijo na predhodno pripravljeni označeni zbirki besedil ali korpusu. Besedila v korpusu ustrezno označi več označevalcev, njihove oznake pa se nato za namene učenja modelov smatrajo kot zlati standard, na podlagi katerega se primerja napovedna učinkovitost modelov.

Težave pri nadzorovanem učenju predstavlja predvsem sestavljanje označenih korpusov, ki je lahko izjemno dolgotrajno in zahteva veliko znanja in izkušenj. Kakovost korpusa namreč neposredno vpliva na klasifikacijske zmožnosti končnega modela. Mozetič, Grčar in Smailović (2016) so opazovali vpliv kakovosti označevanja korpusa na končno uspešnost klasifikacijskih modelov. Zaključili so, da kakovost označevanja korelira z uspešnostjo modelov in predstavlja tudi zgornjo mejo napovedne točnosti modelov. Sama kakovost učnega korpusa je zato enako pomembna kot izbira algoritma za učenje klasifikacijskega modela.

Pred uporabo besedil iz korpusa za učenje modelov je priporočljivo, da se posamezna besedila predhodno obdela. Na ta način se iz vhodnih besedil odstrani informacije, ki niso dovolj informativne za uspešno klasifikacijo, hkrati pa zmanjša dimenzionalnost vhoda in porabo pomnilnika. Uysal in Gunal (2014) sta primerjala vpliv različnih metod predobdelave besedil na končno uspešnost klasifikacijskega modela za turška in angleška besedila. Zaključila sta, da lahko ustrezna predobdelava besedil znatno izboljša klasifikacijske modele in je zato ključen korak v učenju modelov. V nadaljevanju predstavljamo nekaj najpogostejših korakov predobdelave besedil, kot so predstavljene v Rutar (2016).

Tokenizacija je deljenje besedil na posamezne simbole, ki tvorijo besedilo, navadno besede, ločila in številke. Najenostavnejši način tokenizacije je deljenje besedil na presledkih, ki pa zaradi različne stičnosti ločil navadno ni najbolj natančen. Kompleksnejše metode tokeniziranja besedil temeljijo na modelih, ki so predhodno naučeni na večji zbirki besedil posameznega jezika.

Izločevanje neinformativnih besed iz besedil izloči vse besede, ki se sicer zelo pogosto pojavljajo v besedilih, nimajo pa velike informativne vrednosti. Sem spadajo predvsem vezniki (in, ko, ter), glagoli v funkciji pomožnih glagolov (biti) in zaimki (jaz, ti, mi, vi). Včasih se iz besedil izloči tudi druge neinformativne simbole, kot so ločila in številke.

Krnjenje in lematizacija sta metodi, ki se dokaj pogosto uporabljata v jezikih z bogato morfologijo, kot je na primer slovenščina. Pri krnjenju besednim oblikam odstranimo končnice in ohranimo zgolj koren besede, medtem ko pri lematizaciji za vsako besedo poiščemo osnovno obliko besede, kot je navedena v splošnem jezikovnem slovarju. Na ta način zmanjšamo število besed, ki jih mora model pri klasifikaciji obdelati in med njimi razlikovati, pri čemer ohranimo osnovni pomen vsake besede, ki je zajet v korenu.

3.2 Vektorizacija besedil

Klasifikacijski algoritmi kot vhod pričakujejo numerične podatke, zato je potrebno besedilo po predobdelavi pretvoriti v ustrezno strukturirano obliko. Najpreprostejši način preobrazbe besedila v numerično obliko je izdelava binarnih vektorjev dolžine N, kjer je N število vseh različnih besed v korpusu. Vsaki besedi nato priredimo svoj edinstveni vektor, ki ima ničle na vseh mestih, razen na mestu, ki ustreza dani besedi. Besedilo tako predstavimo kot vsoto vektorjev vseh besed, ki besedilo sestavljajo.

Dodatno lahko ta osnovni model izboljšamo tako, da v vektorsko predstavitev besedila vnesemo podatke o pomembnosti posameznih elementov besedila, kar bi omogočilo učinkovitejšo klasifikacijo. Eden izmed takih modelov, ki je zelo priljubljen in se uporablja tudi izven domene analize sentimenta, je tako imenovan TF-IDF model (*term frequeny-inverse document frequency*). Osnovna ideja TF-IDF modela je, da so določene besede v besedilu veliko bolj informativne in bolj pripomorejo k klasifikaciji dokumenta, zato jim lahko dodelimo večjo utež (Manning, Raghavan & Schütze, 2009). Utež, ki jo posamezni besedi dodelimo, je premo sorazmerna s številom pojavitev besede v določenem besedilu v korpusu (TF). Vendar se določene besede, kot so vezniki in zaimki, pogosto pojavljajo v vseh besedilih. Tudi če take nepolnopomenske besede odstranimo v postopku predobdelave, nam lahko zaradi same sestave korpusa še vedno ostanejo določene polnopomenske besede, ki so vseprisotne. V korpusu z avtomobilističnimi besedili je na primer beseda avtomobil zelo pogosta in ne podaja veliko uporabnih informacij za klasifikacijo posameznega dokumenta. Cilj je, da takim besedam prvotno utež ustrezno zmanjšamo, da ne bi preveč vplivala na rezultate klasifikacije. Utež zmanjšamo na podlagi pogostosti posamezne besede v korpusu, kar izračunamo kot:

$$idf_t = log \frac{N}{df_t}$$

pri čemer je N število vseh dokumentov v korpusu, df_t pa število dokumentov, v katerih se pojavi beseda t. Tako je vrednost idf_t velika, če je beseda redka, in nizka, če je beseda pogosta Končna oblika formule modela TF-IDF je:

$$\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t$$

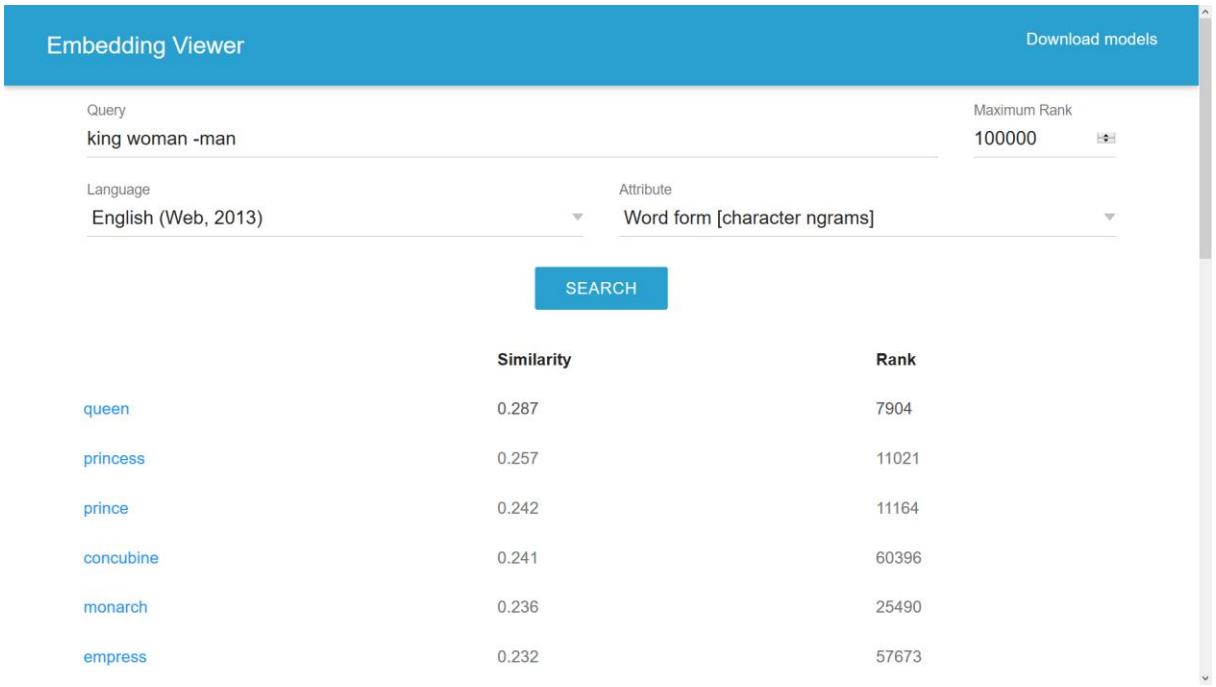
kjer tf predstavlja število pojavitev besede v določenem dokumentu, idf pa obratno vrednost števila dokumentov v korpusu, v katerih se določena beseda pojavi.

Kljub temu se zdi pretirano, da bi beseda, ki se dvajsetkrat pojavlja v določenem dokumentu, res bila dvajsetkrat pomembnejša od besede, ki se pojavi enkrat. Da bi dodatno omilili utež, uporabimo sublinearno obtežitev, pri kateri kot utež namesto pogostosti pojavitve posamezne besede uporabimo njen logaritem:

$$tf_{t,d} = \begin{cases} 1 + \log tf_{t,d} & \text{če je } tf_{t,d} > 0 \\ 0 & drugače \end{cases}$$

TF-IDF model obtežitve lahko prav tako uporabimo na daljših besednih ali celo znakovnih sekvencah in ne zgolj na eni besedi. Te daljše besedne in znakovne sekvence imenujemo n-grami, kjer n označuje dolžino sekvence. Na ta način poskušamo v model vnesti tudi informacije o pomembnosti določenih besednih zvez ali strukture posameznih besed.
Težava take predstavitve besedil je, da besedni vektorji ne nosijo nobene informacije o kakršnikoli povezavi med različnimi besedami. Vzemimo na primer besedišče, ki vsebuje tri besede *hotel*, *motel* in *lenivec*. Vsaki od teh besed priredimo svoj edinstven vektor dimenzije N = 3: *hotel* = [1, 0, 0], *motel* = [0, 1, 0], *lenivec* = [0, 0, 1]. Opazimo lahko, da so vsi trije vektorji ortogonalni in ne izražajo nikakršne podobnosti, čeprav sta v resnici besedi “hotel” in “motel” bolj sorodni kot “lenivec”.

Tako so opisana modela v zadnjih letih pri uporabi določenih klasifikacijskih modelov, predvsem nevronskih mrež, zamenjale vektorske vložitve besed (ang. *word embeddings*), ki so postale standard za vrsto problemov na področju obdelave naravnega jezika. Vektorske vložitve so vektorske predstavitve besed, ki učinkovito zajemajo številne sintaktične in pomenske lastnosti besed (Mikolov, Sutskever, Chen, Corrado & Dean, 2013). Pri preslikavi posameznih besed s tem modelom dobimo besedne vektorje, ki imajo to lastnost, da so si v novem vektorskem prostoru blizu, če so si bile osnovne besede pomensko podobne. Narišimo na primer vektorsko vložitev besede »king«. Nato od te vložitve odštejemo vektorsko vložitev besede »man« in ji prištejmo vektorsko vložitev besede »woman«. Če novo dobljeni vektor narišemo v isto ravnino z vektorskimi vložitvami, opazimo, da leži v bližini vektorske vložitve besede »queen«. Opisani pojav nazorno ponazarja Slika 3.1.



Slika 3.1: Prikaz vektorskih vložitev v orodju Sketch Engine

Vektorske vložitve besed načeloma treniramo na veliki količini neoznačenih podatkov. Dva priljubljena načina treniranja vektorskih vložitev sta modela CBOW (zvezna vreča besed, ang. *continuous bag of words*) in Skip-gram (Mikolov, Chen, Corrado & Dean, 2013). Oba modela predvidevata, da določeno besedo najbolje definira kontekst, v katerem se pojavlja. Tako je pri modelu CBOW cilj treniranja napoved sredinske besede glede na besede v njeni okolici, pri modelu Skip-gram pa napoved okoliških besed glede na sredinsko. Ker je samo treniranje takih vložitev na ogromni količini podatkov zamudno, obstajajo na voljo različne predhodno naučene vektorske vložitve, ki so prosto dostopne. Najbolj znani med njimi so Word2Vec, GloVe in Fasttext.

V zadnjem času so se začele na širšem področju obdelave naravnega jezika pojavljati tudi arhitekture nevronskih mrež, ki kot vhod prejmejo oba načina predstavitve besedil. Martinc in Pollak (2019) sta za klasifikacijo jezikovnih zvrsti razvila arhitekturo, ki združuje samodejno in ročno generirane značilke. Prva vrsta značilk je bila generirana s pomočjo konvolucijskih plasti na podlagi znakovnih vektorskih vložitev, ki so bile izračunane med samim učenjem modela. Ročno generirane značilke so vsebovale predstavitev besedila s TF-IDF in BM25 obtežitvami. Oba modela besedila sta bila izdelana na podlagi znakovnih n-gramov različnih dolžin, obe vrsti značilk pa sta bili nato posredovani v polnopovezano nevronsko mrežo, ki je poskrbela za končno klasifikacijo. Model je bil preverjen na različnih podatkovnih zbirkah za

klasifikacijo jezikovnih zvrsti, pri čemer avtorja poročata, da je izboljšal rezultate dosedanjih modelov. Avtorja sta izvedla tudi primerjavo vpliva posamezne vrste značilk na končno klasifikacijsko uspešnost modela. To sta storila tako, da sta poskušala eksperiment ponoviti s samo eno vrsto značilk in nato rezultate primerjala z rezultati glavne študije. Zaključila sta, da ima TF-IDF obtežitev v izolaciji za njun problem sicer nekoliko večji vpliv na končni rezultat kot samodejno generirane značilke s konvolucijskimi mrežami, vendar kljub temu ne dosega rezultatov, ki jih dosega arhitektura z obema vrstama značilk.

Podobna arhitektura je bila uporabljena v Pelicon, Martinc in Kralj Novak (2019), tokrat na problemu samodejne prepoznave sovražnega govora v tvitih. Razlikovala se je predvsem v tem, da so avtorji za samodejno generiranje značilk namesto konvolucijskih plasti uporabili enosmerne rekurentne nevronske mreže z dolgo-kratkoročno pomnilno celico, pri čemer so rekurentne plasti značilke generirale na podlagi besed. TF-IDF obtežitve so bile generirane tako za znakovne kot besedne n-grame različnih dolžin, končni matriki s TF-IDF obtežitvami pa so bile dodane še nekatere druge ročno generirane značilke, kot so sentiment posameznega tvita, najdaljše zaporedje ločil in število zmerljivk, ki so se pojavljale v posameznem tvitu. Model je na tekmovanju SemEval 2019 v eni od podnalog prepoznavanja sovražnega govora dosegel peto mesto.

3.3 Vnašanje informacije o sentimentu v vložitve

Za učinkovitejše analiziranje sentimenta številni modeli vnašajo dodatne informacije o sentimentu v model. Najpogosteje se to doseže z vnosom podatkov o sentimentu s pomočjo slovarjev sentimenta. V poglavju 2.2 smo predstavili več študij, kjer so bile te informacije vnesene neposredno v matriko s TF-IDF obtežitvami ali drugače ročno dodane kot značilke. Drug pristop je, da podatke o sentimentu vnesemo neposredno v vektorske vložitve besed.

Yu, Wang, Lai in Zhang (2017) navajajo, da vektorske vložitve načeloma ne zajamejo dovolj podatkov o sentimentu, saj nosijo samo informacijo o kontekstu posamezne besede. Tako imamo lahko besede, ki se uporabljajo v zelo podobnih kontekstih, izražajo pa povsem nasproten sentiment. Vzemimo na primer protipomenki *dober* in *slab*. Besedi izražata povsem nasproten sentiment, vendar ker sta obe pridevnika in se pojavljata v zelo podobnih kontekstih, imata zelo podobno vektorsko vložitev. Vnašanja informacije o sentimentu v vektorske vložitve so se lotili tako, da so vzeli javno dostopne predhodno naučene vektorske vložitve in jih na

podlagi informacij iz leksikona sentimenta prilagodili. Pri tem so uporabili leksikon E-ANEW, v katerem je sentiment kodiran z realnim številom na intervalu od 1 do 9, kjer 1 predstavlja najbolj negativen, 5 najbolj nevtralen in 9 najbolj pozitiven sentiment. Za vsako besedo v korpusu so poiskali njenih deset najbližjih sosedov glede na kosinusno razdaljo njihovih vektorskih vložitev. Nato so besede v tej množici razvrstili po sentimentu, in sicer od bolj do manj podobne glede na absolutno razliko v sentimentu iz leksikona sentimenta med besedo iz množice in ciljno besedo. Ko je bil seznam besed sestavljen, so vektorsko vložitev ciljne besede posodobili tako, da je bila bliže vektorskim vložitvam na vrhu seznama, ki so ji bile po sentimentu podobne, in dlje od vektorskih vložitev besedam na dnu seznama, ki so se po sentimentu od ciljne besede razlikovale. Da bi preprečili preveliko spremembo izvirnega vektorskega prostora in posledično izgubo kontekstualnih informacij, so dodali omejitev, ki je določala, koliko se lahko posodobljeni vektor ciljne besede največ razlikuje od izvirnega vektorja. Metodo posodabljanja vektorjev so preizkusili na dveh predhodno naučenih javno dostopnih vektorskih vložitvah Word2vec in GloVe. Posodobljene in izvirne vektorske vložitve so nato uporabili na problemih binarne in petrazredne klasifikacije sentimenta, pri čemer so modele učili na podatkih iz korpusa SST (Socher in drugi, 2013). V primerjavi rezultatov avtorji poročajo o 1,5-odstotni izboljšavi za vektorje GloVe in 1,7-odstotni izboljšavi za vektorje Word2vec na problemu binarne klasifikacije ter o 1,5-odstotni izboljšavi za oboje vektorje na problemu petrazredne klasifikacije.

Tang, Wei, Qin, Yang, Liu in Zhiu (2016) so se vnosa informacij v sentiment v vektorske vložitve lotili tako, da so izdelali svoje vektorske vložitve, ki ne nosijo zgolj informacije o medsebojnem odnosu med besedami, temveč tudi o sentimentu posamezne besede. Za to nalogo so zbrali korpus 5 milijonov tvitov ter vsak tvit označili s sentimentom na podlagi emotikonov. Čeprav tak korpus ni bil enako kakovostno označen, kot če bi ga ročno označevali označevalci, so avtorji predpostavljali, da bo šibka informacija o sentimentu v tako velikem korpusu zadostovala za kakovostne vektorske vložitve. Na podlagi tega korpusa so z nevronskimi mrežami naučili nove vektorske vložitve, kjer je bil cilj naloge, podobno kot pri CBOW modelu, napovedovanje konteksta posamezne besede. Nalogo so dopolnili tako, da je morala mreža hkrati napovedati sentiment tvita, v katerem se je posamezna beseda nahajala. Na ta način so v samo vektorsko vložitev poleg informacije o medsebojni povezavi med besedami vnesli tudi informacijo o sentimentu besede.

Ker je postopek učenja novih vektorskih vložitev dolgotrajen in zahteva velike količine podatkov, smo se v tej diplomski nalogi odločili, da bomo za obogatitev vektorskih vložitev z informacijami o sentimentu preizkusili prvi pristop.

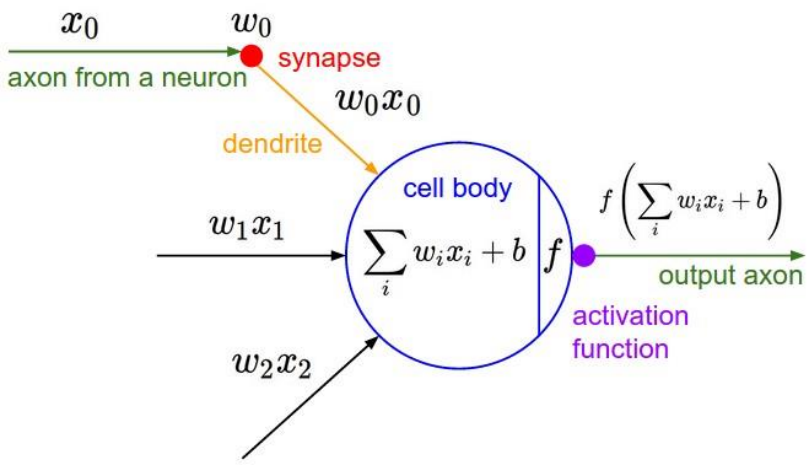
3.4 Nevronske mreže

Nevronska mreža je vrsta klasifikacijskega modela, ki je sestavljen iz enega ali več slojev povezanih vhodno-izhodnih enot ali nevronov, pri čemer je vsaka povezava med nevroni ustrezno obtežena. Sama arhitektura posameznega nevrona se zgleduje po zgradbi nevronov pri sesalcih. V naravi je vsak nevron sestavljen iz dendritov, ki prejmejo vhodne informacije iz okolja ali drugih nevronov, s katerimi so povezani, in jih pošljejo jedru nevrona. Jedro nato vhode ustrezno obdela in jih prek aksona pošlje naprej naslednjemu nevronu.

Podobno si lahko predstavljamo zgradbo umetno ustvarjenega nevrona. Vsak nevron kot vhod prejme atribute posameznega podatka iz podatkovne zbirke oziroma izhodne podatke nevronov iz predhodnega sloja. Vsakemu od teh vhodov je prirejena utež, ki skrbi za vpliv posameznega podatka na končni rezultat. Če je utež velika in pozitivna, bo posamezen vhodni podatek bolj vplival na rezultat in obratno. Vhodnim podatkom je dodana še enota pristranskosti (ang. *bias unit*) s svojo utežjo. Jedro nevrona nato vhodne podatke obdela po naslednji formuli:

$$a = f(\sum_i w_i x_i + b)$$

pri čemer a predstavlja aktivacijo nevrona, x_i predstavlja i-ti vhodni podatek v vhodnem sloju, w_i predstavlja utež i-tega vhodnega podatka, b predstavlja enoto pristranskosti, f pa aktivacijsko funkcijo. Zgradba nevrona je ponazorjena na Sliki 3.2.

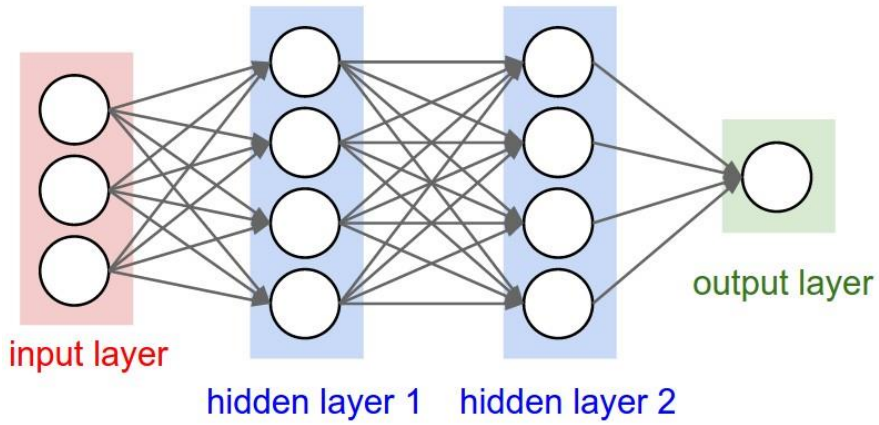


Slika 3.2: Zgradba umetnega nevrona (CS231n Convolutional Neural Network for Visual Recognition, 9. september 2019)

Formalne definicije aktivacijske funkcije ni. Gre za nabor funkcij, ki prejeto vrednost preslikajo v vrednost na določenem intervalu. Ker so ostale operacije v nevronski mreži linearne, aktivacijske funkcije v model vnašajo nelinearnosti. Aktivacijske funkcije so rastoče ter zvezno odvedljive, lastnost, ki omogoča učenje klasifikatorja.

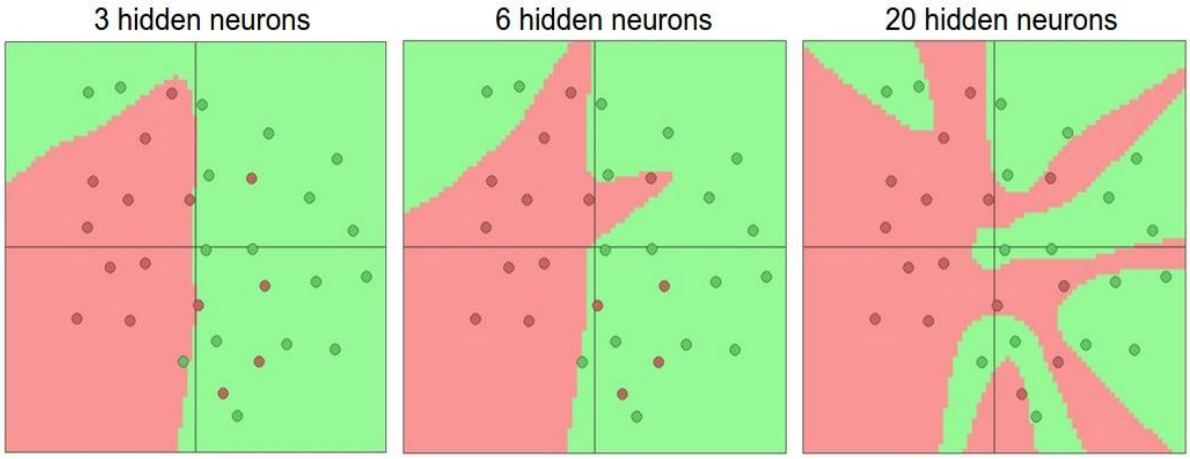
3.4.1 Polnopovezane nevronske mreže

Osnovna oblika nevronskih mrež so polno povezane naprej usmerjene nevronske mreže (ang. *fully-connected feed-forward neural networks*). Modelirane so kot skupek nevronov, ki so povezani v aciklični graf. Nevroni so razdeljeni v več slojev: vhodni sloj, enega ali več skritih slojev in izhodni sloj. Vhodni sloj predstavlja vhodne atribute posameznega podatka, katerim je dodana enota pristranskosti. Ti nevroni se nato povezujejo v naslednji, skriti sloj, ki vhodne podatke obdela in jih pošlje naprej v naslednji skriti sloj ali v izhodni sloj. Izhodni sloj nato vrne rezultat klasifikacije. Omeniti je treba, da je v tej arhitekturi vsak nevron povezan natančno z vsemi nevroni naslednjega sloja in z nobenim nevronom v istem sloju oziroma predhodnih slojih. Arhitekturo nevronskih mrež in postopek učenja povzemamo po knjigi *Neural Networks and Deep Learning* (Nielsen, 2015). Primer arhitekture dvoslojne polnopovezane nevronske mreže je predstavljen na Sliki 3.3.



Slika 3.3: Arhitektura dvoslojne nevronske mreže (CS231n Convolutional Neural Network for Visual Recognition, 9. september 2019)

Mogoče je pokazati, da je enoslojna nevronska mreža, tj. nevronska mreža z enim skritim slojem, univerzalni aproksimator za zvezne funkcije. To pomeni, da za vsako zvezno funkcijo obstaja enoslojna nevronska mreža, ki lahko dano funkcijo aproksimira. Slika 3.4 prikazuje tri grafe funkcij, ki so rezultat enoslojne nevronske mreže z različnim številom nevronov v skritem sloju. Kljub temu kot klasifikatorje uporabljamo nevronske mreže z več skritimi sloji. V praksi se namreč pokaže, da lahko z večanjem števila slojev ter nevronov v posameznem sloju nevronska mreža potencialno aproksimira večje število nelinearnih funkcij, ki so za naše namene klasifikacije zanimivejše. Število slojev in nevronov v vsakem skritem sloju sta prosta parametra, ki jih je potrebno prilagoditi glede na posamezen problem. Imenujemo ju hiperparametra algoritma.



Slika 3.4: Primer funkcij, ki so rezultat enoslojne nevronske mreže s tremi, šestimi in dvajsetimi nevroni v skritem sloju (CS231n Convolutional Neural Network for Visual Recognition, 9. september 2019)

3.4.2 Učenje nevronske mreže

Za uspešno klasifikacijo je model nevronske mreže treba naučiti na učni množici. Učenje nevronske mreže zajema prilagajanje uteži in enot pristranskosti na povezavah do nevronov, da bi dosegli čim boljše ujemanje med napovedmi nevronske mreže in podatki v učni množici.
Pri inicializaciji nevronske mreže uteži inicializiramo na naključne vrednosti. Kakovost kompleta uteži, ki so v uporabi, se kaže pri uspešnosti klasifikacije našega modela. Seveda je praktično nemogoče, da bi model z naključno inicializiranimi utežmi bil izjemno uspešen pri klasifikaciji. Zato želimo kakovost posameznega kompleta uteži, ki je trenutno v uporabi, tudi dejansko oceniti, da lahko na podlagi objektivne metrike uteži kasneje prilagodimo. V ta namen uporabljamo cenitveno funkcijo C, s katero izračunamo razliko med rezultatom klasifikatorja in dejanskim razrednim atributom podatka.

Pri učenju želimo uteži na povezavah nevronov prilagoditi tako, da cenitveno funkcijo C minimiziramo, saj želimo na ta način čimbolj zmanjšati napako klasifikatorja. Metoda gradientnega spusta (ang. *gradient descent*) cenitveno funkcijo minimizira tako, da sledi njenemu odvodu. S pomočjo odvoda lahko namreč uteži spreminjamo tako, da se približamo minimumu cenitvene funkcije. Tako v vsakem koraku metode izračunamo vektor delnih odvodov cenitvene funkcije C po vseh njenih utežeh oziroma gradient. Uteži nato posodobimo tako, da parameter α, ki predstavlja hitrost učenja, pomnožimo z dobljenimi delnimi odvodi in dobljeno vrednost odštejemo od uteži:

$$\Delta w^{(k)} = \Delta w^{(k-1)} - \alpha * G$$

pri čemer Δw predstavlja vektor uteži v k-tem koraku metode, G pa gradient.

Hitrost učenja α je parameter, ki določa velikost premika v smeri odvoda. Sama izbira vrednosti α je zelo občutljiva, saj vpliva na hitrost in učinkovitost učenja. Če je vrednost premajhna, se bo vrednost cenitvene funkcije sicer pomikala proti 0, vendar bo morda za minimizacijo potrebno korake gradientnega spusta velikokrat ponoviti, medtem ko lahko pri preveliki vrednosti α vrednost cenitvene funkcije sploh ne konvergira proti 0.

Gradientni spust načeloma nadaljujemo, dokler ne dosežemo minimuma cenitvene funkcije. Vendar se v praksi to izkaže za potencialno dolgotrajen postopek. Samo učenje se proti koncu začne namreč upočasnjevati, ker so odvodi pri minimumu funkcije vedno manjši. Zato za gradientni spust načeloma nastavimo zaustavitveni kriterij. To je lahko število ponovitev postopka ali epoh, po katerem postopek zaustavimo, ali dovolj majhna vrednost cenitvene funkcije.

Izračun gradienta uteži nevronske mreže izračunamo z algoritmom vzvratnega razširjanja napake (ang. *backpropagation*). To storimo tako, da najprej izračunamo klasifikacijsko napako na izhodnem sloju nevronske mreže. Slednja je definirana na naslednji način:

$$\delta^L_j = \partial C/\partial a^L_j \ * \sigma'(z^L_j)$$

kjer je $\partial C/\partial a^L_j$ odvod cenitvene funkcije po j-tem nevronu izhodnega sloja, $\sigma'(z^L_j)$ pa odvod aktivacije j-tega nevrona v izhodnem sloju. Formulo lahko zapišemo v matrični obliki kot:

$$\delta^L = \nabla aC \odot \sigma'(z^L)$$

kjer člen ∇aC predstavlja vektor delnih odvodov, $\sigma'(z^L)$ vektor odvodov aktivacijske funkcije, ⊙ pa hadamardov produkt med vektorjema. S pomočjo napake enega sloja lahko nato izračunamo napako predhodnega sloja, vse dokler ne pridemo do prvega sloja mreže. Napako l-tega sloja izračunamo po formuli:

$$\delta^l = ((w^{l+1})^T * \delta^{l+1}) \odot \sigma'(z^l)$$

kjer je $(w^{l+1})^T$ transponirani vektor uteži naslednjega sloja. Napake na posameznih nevronih so povezane z delnimi odvodi cenitvene funkcije C po utežeh in enoti pristranskosti, kar prikazujeta naslednji enačbi:

$$\partial C/\partial w^l_{jk} = a^{l-1}_k * \delta^l_j$$

kjer indeks l predstavlja sloj, indeks j pa nevron in

$$\partial C/\partial b^l_j = \delta^l_j$$

torej je napaka j-tega nevrona v l-tem sloju kar enaka odvodu cenitvene funkcije po enoti pristranskosti.

Strnimo sedaj postopek algoritma vzvratnega širjenja napake. Najprej podamo podatek iz učne množice modelu kot vhodni podatek. Nato izračunamo napako med dejanskim razrednim atributom in napovedanim razrednim atributom na izhodnem sloju ter jo razširimo *nazaj* po slojih mreže. Na ta način dobimo napake nevronov na vseh slojih, s pomočjo katerih lahko izračunamo odvode uteži. Ta postopek nato ponovimo za vse podatke v učni množici.

3.4.3 Rekurentne nevronske mreže

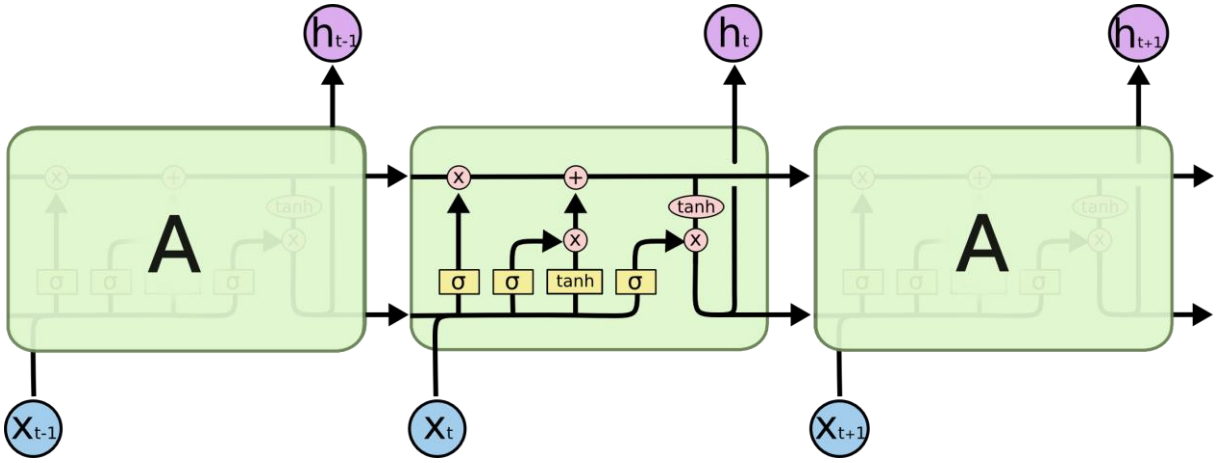
Besede v besedilu ne nosijo pomena same po sebi, temveč ga tvorijo v kombinaciji z drugimi besedami, ki se pojavljajo v istih povedih in istem besedilu. Besedilo lahko tako razumemo kot sekvenco različnih simbolov, ki skupaj posredujejo neko informacijo. Da bi lahko besedilo modelirali kot sekvenco, so raziskovalci oblikovali varianto nevronske mreže, imenovano rekurentna nevronska mreža.

Rekurentna nevronska mreža v osnovi deluje enako kot polnopovezana mreža s to razliko, da vhodno besedilo obdela v več korakih, pri čemer izhodno informacijo iz prejšnjega koraka uporabi kot informacijo pri izračunu izhoda v trenutnem koraku. V primeru besedila mreža v vsakem koraku obdela en simbol ter izhodno informacijo uporabi pri obdelavi naslednjega simbola in tako vse do konca besedila. Matematično je operacija opredeljena kot:

$$h_t = \emptyset(Wx_t + Uh_{t-1})$$

kjer h_t predstavlja izhodno informacijo oziroma skrito predstavitev v trenutnem koraku, W predstavlja matriko uteži, x predstavlja vhod, U predstavlja matriko uteži za skrito predstavitev prejšnjega koraka, h_{t-1} skrito predstavitev prejšnjega koraka, $\emptyset$ pa aktivacijsko funkcijo. Skrite predstavitve tako vsebujejo sledi informacij iz vseh predhodnih korakov, tako da bo naša posodobljena oblika mreže pri klasifikaciji upoštevala vse besede v vrstnem redu, kot se pojavljajo v besedilu.

Učenje rekurentnih mrež poteka na enak način kot učenje polnopovezane nevronske mreže s to razliko, da mora sedaj gradient teči skozi vse korake obdelave. Posodobljeni učni algoritem imenujemo vzvratno razširjanje napake skozi čas. Ker je vsak korak pravzaprav sam po sebi nevronska mreža, lahko pri učenju rekurentnih nevronskih mrež hitro pride do izjemno velikih ali izjemno majhnih gradientov. To povzroči, da določen signal preveč vpliva na obdelavo oziroma oslabi zelo zgodaj v postopku učenja. Ta pomanjkljivost rekurentnih nevronskih mrež je bila omiljena z uvedbo dolgo-kratkoročne pomnilne celice (ang. *long-short term memory* ali *LSTM*). Vsak korak obdelave vsebuje svojo pomnilno celico, ki vsebuje informacije ločeno od skritih predstavitev. Pomnilna celica je sestavljena iz štirih vhodno-izhodnih vrat. Vrata so funkcije, ki določajo, koliko informacij se bo zapisalo v trenutno skrito predstavitev in koliko informacij se bo poslalo v pomnilno celico v naslednjem koraku obdelave. Na ta način ostaja napaka pri vzvratnem razširjanju v mejah, ki omogočajo, da se celotna mreža učinkoviteje uči. Nevron s pomnilno celico LSTM je grafično predstavljen na Sliki 3.5, kjer so vrata označena z rumenimi pravokotniki. Za podrobnejši in matematično popolnejši pregled mrež s pomnilnimi celicami LSTM bralcu predlagamo članek avtorjev Greff, Srivastava, Koutnik, Steunebrink in Schmidhuber (2017).



Slika 3.5: Prikaz treh pomnilnih celic LSTM (Colah's blog, 2015)

4 METODOLOŠKI PRISTOP

V tem poglavju predstavimo metodološki pristop, ki smo ga uporabili za izvedbo eksperimenta. Najprej opišemo podatkovno množico, ki smo jo uporabili za učenje modelov in njihovo evalvacijo. Nato opišemo arhitekturo nevronskih mrež, ki smo jo uporabili za naše modele. Nato opišemo postopek učenja vseh petih modelov analize sentimenta v novicah. Poglavje zaključimo z opisom postopka validacije naučenih modelov.

4.1 Opis podatkov

Za namene razvoja modela za zaznavanje sentimenta v slovenskih novicah smo uporabili korpus slovenskih novic SentiNews (Bučar, Žnidaršič & Povh, 2018). Korpus je sestavljen iz 10.427 novic z ekonomsko, finančno in politično vsebino, ki so bile zbrane na petih najbolj branih slovenskih spletnih novinarskih portalih, in sicer 24ur, Dnevnik, Finance, Rtvslo in Žurnal24, v obdobju med 1. septembrom 2007 in 31. decembrom 2013. Novice je šest označevalcev ročno označilo na treh stopnjah granularnosti, in sicer na nivoju besed, povedi in celotnega dokumenta. Med označevanjem so morali označevalci sentiment v novicah označiti s stališča povprečnega slovenskega bralca, pri čemer so odgovorili na vprašanje, kako so se po branju novice počutili. Označevalci so ocene podajali na podlagi petstopenjske Lickterjeve lestvice, pri čemer predstavlja ocena 1 najbolj negativno oceno, ocena 5 pa najbolj pozitivno. Končna ocena je bila določena kot povprečje ocen vseh označevalcev. Petstopenjska lestvica je bila nato preslikana v tri opisne kategorija sentimenta, negativno, nevtralno in pozitivno, pri čemer so bile novice s povprečno oceno do vključno 2,4 označene kot negativne, novice s povprečno oceno od 2,4 do 3,6 kot nevtralne in novice s povprečno oceno od vključno 5 kot pozitivne. Končni označeni del korpusa tako vsebuje 3337 negativnih, 5425 nevtralnih in 1665 novic, kar kaže na dokaj drastično neuravnoteženost kategorij.

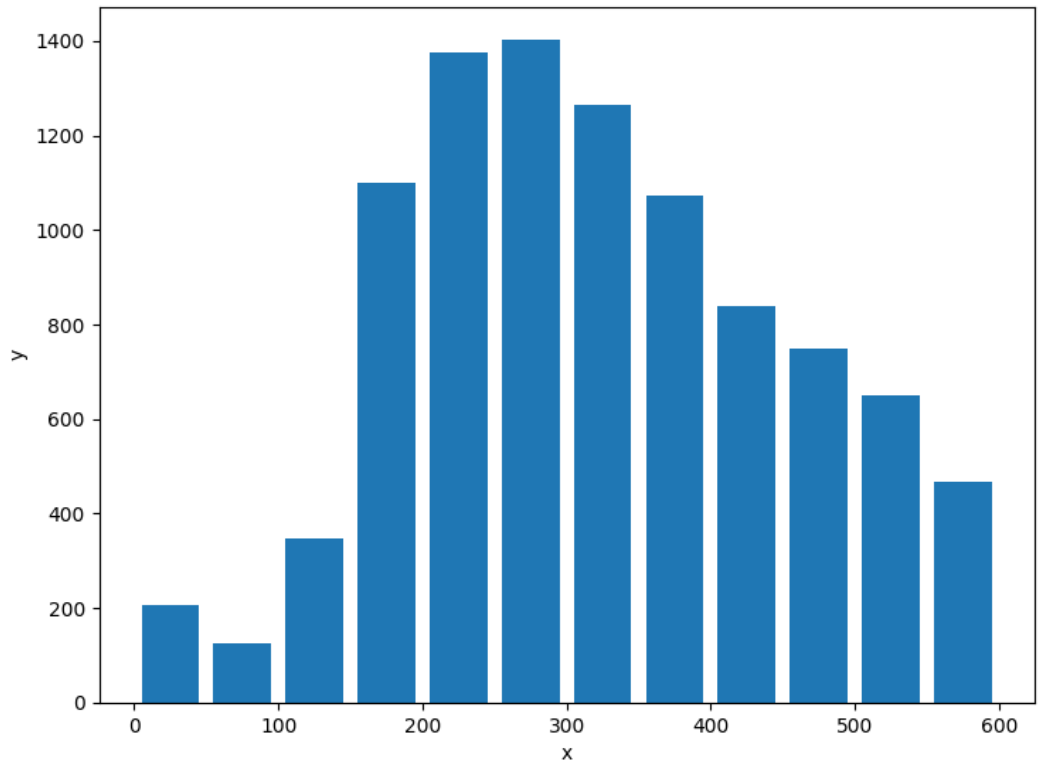
Avtorji so kakovost označevanja in konsistentnost označevalcev ocenili s štirimi merami strinjanja, in sicer Cronbachovo alfo, Krippendorfovo alfo, Fleissovo kapo in Kendallovim koeficientom konkordance. Ker so bile mere strinjanja najvišje pri oznakah na ravni dokumenta,

smo se odločili, da bomo za namene nadzorovanega učenja modela uporabili oznake na tej ravni.

4.2 Opis arhitekture

Za modeliranje sentimenta novic uporabimo klasifikacijski model na osnovi nevronskih mrež. Model kot vhod tako prejme besedilo kot celoto in ga v nadaljevanju obdela v dveh obdelovalnih cevovodih. V prvem obdelovalnem cevovodu smo vsako novico najprej tokenizirali. Nato smo vsako besedo, ki se pojavi v korpusu, preslikali v ustrezno vektorsko vložitev. V našem primeru smo uporabili predhodno naučene 300 dimenzionalne vektorske vložitve Fasttext (Grave, Bojanowski, Gupta, Joulin & Mikolov, 2018), ki so jih razvili v okviru razvojnega oddelka podjetja Facebook. Tako smo dobili matriko velikosti R^{dxm}, kjer je d število besed v novici, m pa dimenzija vektorske vložitve.

Novice v korpusu so različne dolžine, mi pa želimo model učiti tako, da mu za vsako posodobitev uteži posredujemo manjše serije dokumentov iz korpusa. Za namene časovno učinkovite vektorske implementacije takega modela moramo rekurentni nevronski mreži posredovati vhodne podatke enake dolžine. Zato smo posamezne novice po potrebi skrajšali ali podaljšali. Posamezno novico smo podaljšali tako, da smo konec matrike zapolnili z ničelnimi vektorji, ki ne predstavljajo besed. Dolžino novic smo glede na distribucijo dolžin novic v korpusu, predstavljeni na Sliki 4.1, določili na 500 besed. Na ta način smo zmanjšali dimenzionalnost modela in porabo pomnilnika ter pohitrili učenje, hkrati pa preprečili, da bi izgubili preveč informacij iz korpusa.



Slika 4.1: Distribucija dolžin besedil v korpusu glede na število besed

Predobdelane novice smo tako poslali v rekurentni sloj z LSTM celicami, pri čemer je vsak izmed slojev vhod preslikal v vektor dimenzije 120, ki je predstavljal samodejno generirano predstavitev podane sekvence. Predvsem pri modelih z velikim številom parametrov se pogosto zgodi, da se mreža preveč prilagodi podanim podatkom in se tako ne nauči splošnih vzorcev iz podatkovne množice (ang. *overfitting*). Da bi preprečili ta pojav med učenjem, smo kot regularizacijsko metodo uporabili izpust (ang. *dropout*). Izpust deluje tako, da med vsakim korakom učenja vsak nevron in vse njegove povezave z neko verjetnostjo p izpustimo (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). Vrednost p je hiperparameter, ki ga določimo pred začetkom učenja. Pri rekurentnih nevronskih mrežah se izpust lahko uporabi neposredno na nevronih v vsakem časovnem koraku, lahko pa tudi na rekurentnih povezavah, ki nevronom posredujejo informacijo iz predhodnega koraka (Moon, Choi, Lee & Song, 2015). V našem modelu smo uporabili obe vrsti izpusta, ki smo ju nastavili na 0,4.

Da bi med samodejno generiranimi notranjimi predstavitvami besedila pridobili samo tiste, ki najbolje predstavljajo vhodno besedilo, smo izhod iz rekurentnega sloja poslali še skozi sloj globalne izbire maksimalnih vrednosti (ang. *global max pooling*). Ta sloj prejme kot vhod celoten vektor in vrne zgolj element z najvišjo vrednostjo. Na ta način zmanjšamo dimenzionalnost vhoda pred naslednjimi sloji v arhitekturi, tako da izberemo zgolj najbolj diskriminativne značilke.

Tudi v drugem obdelovalnem cevovodu smo vsako vhodno besedilo iz učne množice najprej tokenizirali. Nato smo izdelali dve matriki s TF-IDF obtežitvami. V prvi matriki obtežimo vse besedne N-grame dolžine od 1 do 5, pri čemer dodamo omejitev, da v končno matriko vključimo zgolj tiste N-grame, ki se v učni množici pojavijo vsaj petkrat. Na tak način nekoliko zmanjšamo dimenzionalnost matrike, pri čemer predpostavljamo, da tak poseg odstrani vse izjemno redke besede in besedne zveze, ki ne dodajo veliko teže pri klasifikaciji, ter besede in besedne zveze, ki so morda zatipkane. Vse tako pridobljene N-grame tudi obtežimo s sublinearno obtežitvijo.

V drugi matriki obtežimo s TF-IDF utežmi vse N-grame, sestavljene iz znakov dolžine od 1 do 7. Tudi v tem primeru iz končne matrike odstranimo vse N-grame, ki se pojavljajo manj kot 5-krat, preostale N-grame pa obtežimo s sublinearno obtežitvijo. Nato obe matriki združimo v eno TF-IDF matriko dimenzije R^{nxm}, kjer n predstavlja število besedil v učni množici, m pa število značilk.

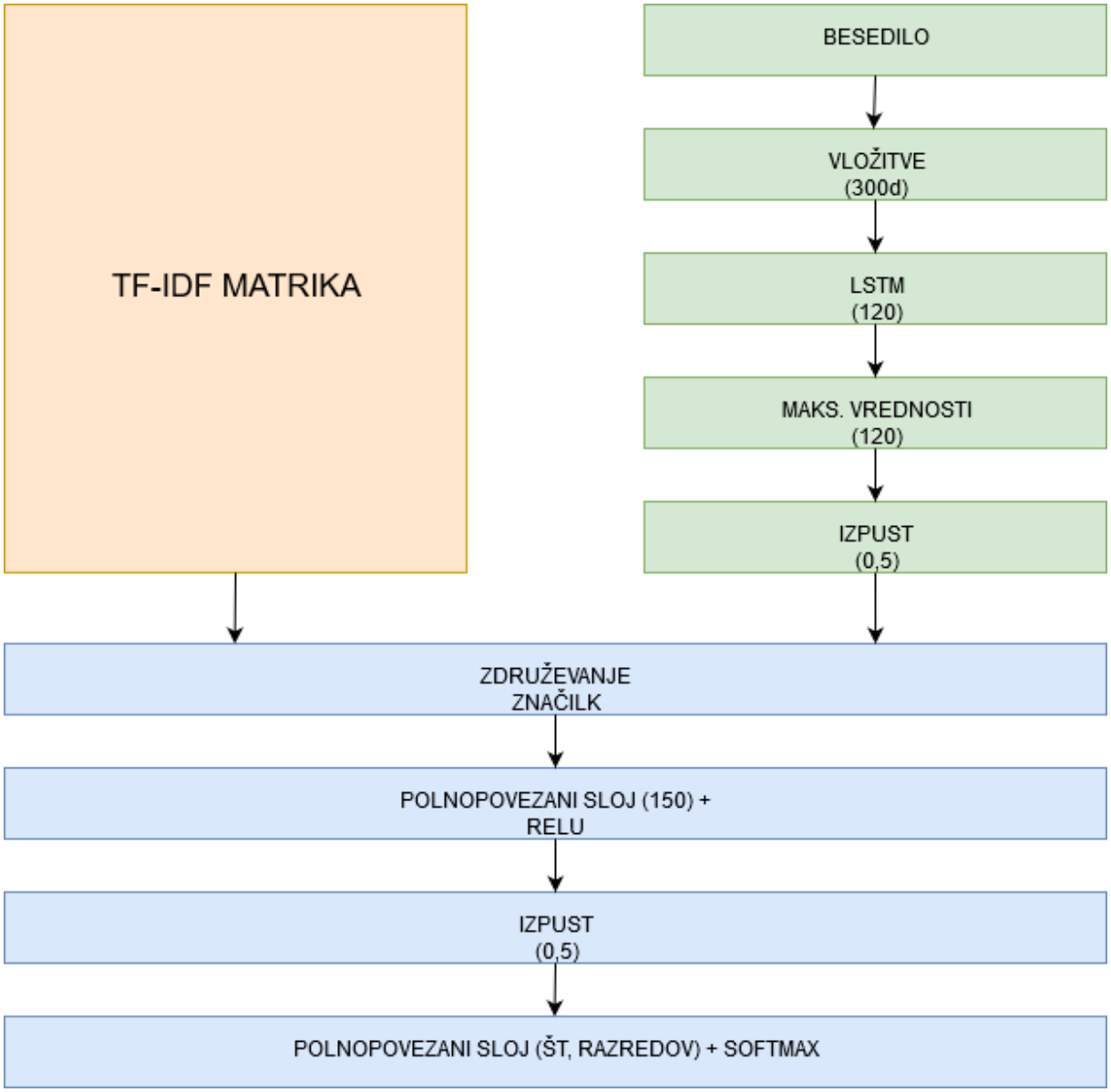
Matriki, ki jih pridobimo iz obeh obdelovalnih cevovodov, nato združimo in pošljemo v polnopovezano nevronsko mrežo, ki skrbi za končno klasifikacijo. Mreža je sestavljena iz ene skrite plasti in ene izhodne plasti. Skrita plast vsebuje 150 nevronov, kot aktivacijska funkcija pa je uporabljena funkcija Relu, ki je definirana kot:

$$f(x) = \max(0, x)$$

Kot regularizacijsko metodo smo tudi na tem sloju uporabili izpust z vrednostjo parametra p 0,5. Izhodno plast sestavljajo trije nevroni, torej po en nevron za vsako ciljno oznako. Na izhodnem sloju uporabimo aktivacijsko funkcijo softmax, ki vrne verjetnostno porazdelitev vhodnega parametra glede na ciljne oznake in je definirana kot:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=0} e^{x_j}}$$

Kot napovedani razred nato vzamemo tisto ciljno oznako z največjo verjetnostjo. Za bolj celovit pregled je opisana arhitektura grafično predstavljena na Sliki 4.2.



Slika 4.2: Arhitektura modelov nevronskih mrež

4.3 Opis učenja modelov

Algoritmi za učenje nevronskih mrež so stohastični in na njihovo učinkovitost lahko pomembno vpliva začetna naključna inicializacija parametrov. Da bi zagotovili ponovljivost poizkusov in

omogočili primerljivost naučenih modelov, smo tako začetku učenja funkcijam za generiranje naključnih števil določili seme, tako da so bili začetni parametri deterministično določeni.

Korpus podatkov smo najprej naključno razdelili na učno in testno množico v razmerju 80 : 20. Nato smo kot učno metodo uporabili stohastični gradientni spust s podmnožicami učne množice (ang. *stochasitc minibatch gradient descent*). Metoda je ena od variant gradientnega spusta, ki zagotavlja dobro konvergiranje učenja ob manjši porabi pomnilnika (Ruder, 2017). Osnovni gradientni spust namreč računa odstopanja za celotno učno množico, preden posodobi uteži modela. To za večje množice podatkov ni primeren način učenja, saj nimamo na voljo dovolj pomnilnika, da bi vanj naenkrat naložili celotno podatkovno zbirko. Gradientni spust s podmnožicami učne množice za razliko od osnovne metode deluje tako, da v vsakem koraku učenja modelu posreduje N podatkov iz učne množice, kar zmanjša porabo pomnilnika. Parameter N je hiperparameter učnega modela, ki ga je potrebno previdno izbrati, saj lahko pri premajhnih N-jih gradient preveč niha, kar onemogoča, da bi model dosegel minimum. V našem primeru smo hiperparameter N določili na 32, saj smo želeli doseči čim boljše razmerje med učinkovitostjo učenja in porabo pomnilnika. Kot cenitveno funkcijo, ki je merila odstopanja med napovedanimi in pravimi oznakami na izhodu, smo uporabili kategorično križno entropijo, ki je definirana kot:

$$C = -\sum_{i}^{M} t_i \log(s_i)$$

kjer t_i in s_i predstavljata prave in napovedane oznake za vsak razred M.

Cilj učenja je bil minimizirati navedeno cenitveno funkcijo, kar posledično zmanjša razliko med napovedanimi in pravimi oznakami. Po vsakem koraku smo glede na izračunano odstopanje med napovedanimi in pravimi oznakami v trenutni podmnožici podatkov uteži modela prilagodili z algoritmom ADAM (Kingma & Ba, 2017), pri čemer je bila začetna učna stopnja nastavljena na 0,001.

Na zgoraj opisani način smo naučili pet modelov, ki si vsi delijo arhitekturo iz poglavja 4.2, razlikujejo pa se v predobdelavi vhodnih podatkov in uporabljenih vektorskih vložitvah besed. Pri modelu LSTM_BOW smo vhodna besedila minimalno obdelali, in sicer smo zgolj vse

velike črke v celotni novici spremenili v male. Za preslikavo besedila v številske vektorje smo uporabili predhodno naučene 300-dimenzionalne slovenske vektorske vložitve Fasttext (Grave, Bojanowski, Gupta, Joulin & Mikolov, 2018), ki so jih razvili v okviru razvojnega oddelka podjetja Facebook. Pri modelu LSTM_BOW+TPROC so bila vhodna besedila nekoliko temeljiteje predobdelana, in sicer smo iz besedil odstranili vse številke, neinformativne besede in ločila. Krnjenja kot metode za predobdelave besedil nismo uporabljali. Pri modelih LSTM_BOW+REF in LSTM_BOW+TPROC+REF smo vektorske vložitve Fasttext na podlagi metode iz poglavja 3.3 popravili z informacijami o sentimentu besed. Sentimente slovenskih besed smo pridobili v leksikonu sentimenta JOB (Bučar, Žnidaršič & Povh, 2018). Leksikon JOB je bil sestavljen na istem korpusu kot smo ga uporabili za učenje našega modela. Vendar so informacije iz leksikona bile pridobljene s pomočjo oznak na nivoju posameznih povedi besedil v korpusu, medtem ko smo za učenje naših modelov uporabili oznake na nivoju besedil, ko je omenjeno v poglavju 4.1. Modela se razlikujeta po predobdelavi besedil, in sicer je bila za model LSTM_BOW+TPROC+REF uporabljena temeljitejša predobdelava, ki je bila uporabljena tudi za model LSTM_BOW+TPROC. Pri zadnjem modelu LSTM_BOW+REF2 so bila besedila enako minimalno obdelana kot pri modelu LSTM_BOW, uporabljene pa so bile popravljene vektorske vložitve Fasttext. Za razliko od ostalih modelov, kjer se vektorske vložitve med učenjem niso posodabljale, smo jih v tem primeru med samim učenjem modela dodatno prilagajali. S tem smo upali, da bodo na podlagi podatkov iz učnega korpusa zajele dodatne informacije, ki so specifične za podani problem, kar bi pozitivno vplivalo na zmogljivost modela. Za nazornejšo predstavitev so vse konfiguracije modelov zbrane v Tabeli 4.1.

Modele smo implementirali v programskem jeziku Python s pomočjo knjižnic Keras in Tensorflow (Abadi in drugi, 2015), ki omogočata učinkovito paralelno učenje mrež na grafičnih karticah. Vse modele smo trenirali 10 epoh, pri čemer smo v vsaki epohi model učili na celotni učni množici. Modele smo učili s pomočjo grafične kartice Nvidia Geforce RTX 2080 z 8 GB delovnega pomnilnika.

Tabela 4.1: Opis modelov nevronskih mrež, ki so bili naučeni v sklopu diplomske naloge

Model	**Predobdelava besedila**	**Vektorske vložitve**
LSTM_BOW	Sprememba velikih črk v male	300-dimenzionalne slovenske prednaučene vložitve Fasttext
LSTM_BOW+TPROC	Sprememba velikih črk v male, odstranitev številk, neinformativnih besed in ločil	300-dimenzionalne slovenske prednaučene vložitve Fasttext
LSTM_BOW+REF	Sprememba velikih črk v male	300-dimenzionalne slovenske prednaučene vložitve Fasttext, prilagojene glede na sentiment
LSTM_BOW+TPROC+REF	Sprememba velikih črk v male, odstranitev številk, neinformativnih besed in ločil	300-dimenzionalne slovenske prednaučene vložitve Fasttext, prilagojene glede na sentiment
LSTM_BOM+REF2	Sprememba velikih črk v male	300-dimenzionalne slovenske prednaučene vložitve Fasttext, prilagojene glede na sentiment in problem

4.4 Validacija modelov

Po koncu učenja smo ocenili uspešnost modela na testni množici, ki je vsebovala tiste podatke iz korpusa, ki jih model med učenjem ni videl. Modelu smo posredovali testno množico kot vhod, kot izhod pa nam je model vrnil napovedane oznake za vsako besedilo v testni množici. Napovedi modela smo primerjali s pravimi oznakami in jih ovrednotili s petimi merami, ki merijo skladnost napovedi modela z oznakami testne množice, in sicer klasifikacijsko točnostjo, natančnostjo, priklicem, oceno F1 in Krippendorfovo alfo. Za boljšo vizualno predstavitev uspešnosti modela in lažji izračun mer uspešnosti smo napovedi najprej predstavili v matriki zmot. V matriki zmot prikažemo število ujemanj med napovedmi modela in oznakami v zlatem

standardu. Na Sliki 4.2 je prikazan primer matrike zmot za naš trirazredni klasifikacijski problem. Opazimo lahko, da so napovedi modela, ki sovpadajo s pravimi oznakami, razporejene po diagonali matrike.

Tabela 4.2: Primer matrike zmot za naš trirazredni klasifikacijski problem

		Napovedani razredi		
		Negativni	Nevtralni	Pozitivni
Resnični razredi	Negativni			
	Nevtralni			
	Pozitivni			

Klasifikacijska točnost je standardna mera za ocenjevanje zmogljivosti modelov strojnega učenja. Definirana je kot delež pravilno klasificiranih primerov. Klasifikacijska točnost je preprosta mera, ki pa ima pomanjkljivosti, ko jo uporabljamo pri ocenjevanju zmogljivosti modelov na testnih množicah z zelo neuravnoteženimi razredi. Vzemimo ekstremen primer take množice z dvema razredoma A in B, kjer 91 primerov spada v razred A, 9 pa v razred B. Na taki množici naučimo model, ki ne diskriminira med razredoma, temveč vse primere privzeto klasificira v razred A. Kljub temu, da tak model ni sposoben prepoznati primerov, ki spadajo v razred B, je njegova klasifikacijska točnost 91-odstotna.

Ker so razredi v uporabljeni podatkovni množici dokaj neuravnoteženi, smo za natančnejšo evalvacijo modela uporabili dodatne mere. Poleg klasifikacijske točnosti sta široko uporabljeni meri natančnost (ang. *precision*) in priklic (ang. *recall*), ki sta definirani kot:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Obe meri se za razliko od klasifikacijske točnosti računata za posamezen razred. Razred, za katerega meri računamo, označimo kot pozitiven razred, ostale razrede pa označimo kot negativen razred. Za pozitiven razred je natančnost definirana kot delež pravilno klasificiranih

pozitivnih primerov (TP) izmed vseh primerov, ki jih model klasificira kot pozitivne (TP + FP). Priklic je za pozitiven razred definiran kot delež pravilno klasificiranih pozitivnih primerov (TP) izmed vseh primerov, ki so v zlatem standardu označeni kot pozitivni (TP + FN).

V praksi se izkaže, da sta natančnost in priklic pogosto obratno sorazmerni in ju je težko istočasno maksimirati. Zato se kot bolj uravnoteženo metriko uporablja oceno F1. Ocena F1 predstavlja harmonično povprečje natančnosti in priklica in je definirana kot:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Tudi ocena F1 se računa za vsak razred posebej, zato se pri večrazrednih problemih kot končno mero uporabi povprečje ocen F1, ki so bile izračunane za vsak razred posebej.

Kot zadnjo metriko za oceno učinkovitosti modelov bomo uporabili Krippendorfovo alfo. Ta metrika ni standardna mera za ocenjevanje uspešnosti modelov, temveč se običajno uporablja za preverjanje skladnosti označevanja med različnim številom označevalcev. Za uporabo alfe v našem primeru lahko po vzoru iz Mozetič, Grčar in Smailović (2016) predpostavimo, da je en označevalec naš naučen model, drugega označevalca pa predstavljajo oznake v zlatem standardu.

Krippendorfova alfa (Krippendorff, 2011) je opredeljena kot:

$$\alpha = 1 - \frac{D_o}{D_e}$$

kjer D_o predstavlja dejansko nestrinjanje med označevalci, D_e pa pričakovano nestrinjanje po naključju. Meri nestrinjanja sta nadalje opredeljeni kot:

$$D_o = \frac{1}{n} \sum_c \sum_k o_{ck} \delta_{ck}^2$$

$$D_e = \frac{1}{n(n-1)} \sum_c \sum_k n_c \times n_k \, \delta_{ck}^2$$

Argumenti o_{ck}, n_c, n_k in n predstavljajo vrednosti v koincidenčni matriki, ki bo predstavljena v nadaljevanju. δ_{ck} je diferenčna funkcija med dvema oznakama in je definirana različno glede na tip oznak. Oznake 'negativno', 'nevtralno' in 'pozitivno' so diskretne in za dani problem jih razumemo kot urejene ter jim pripišemo urejene vrednosti, in sicer oznaki *negativno* -1, oznaki *nevtralno* 0 in oznaki *pozitivno* 1. Diferenčno funkcijo nato definiramo kot:

$$\delta(c,k) = |c-k| \;\; c,k \in \{-1,0,1\}$$

Opazimo lahko, da dana diferenčna funkcija poda razliko 1 med dvema sosednjima razredom, negativnim in nevtralnim ter pozitivnim in nevtralnim, in razliko 2 med dvema skrajnima razredoma. Na ta način napake med skrajnima razredoma štejemo kot hujše v primerjavi z napakami med sosednjima razredoma, česar druge uporabljene mere niso upoštevale.

Za izdelavo koincidenčne matrike moramo najprej predstaviti oznake naših dveh označevalcev za vsak primer v testni množici. To storimo v tako imenovani zanesljivostni matriki, kjer vsaka vrstica predstavlja enega označevalca, vsak stolpec pa enoto, ki sta jo označevalca ocenila. Primer zanesljivostne matrike je predstavljen v Tabeli 4.3.

Tabela 4.3: Primer zanesljivostne matrike za naš problem

Primeri	Besedilo 1	Besedilo 2	...	Besedilo N
Model				
Zlati standard				

Podatke iz zanesljivostne matrike nato predstavimo v koincidenčni matriki (glej Tabelo 4.4), v kateri so na oseh c in k predstavljene vse oznake, s katerimi sta označevalca lahko označila posamezne primere. Vsak vnos v matriki predstavlja število primerov, za katere sta označevalca podala oceni iz stolpca in vrstice. Opazimo lahko, da je število vseh parov oznak v koincidenčno matriko vneseno dvakrat, enkrat kot par c-k, enkrat kot par k-c. Koincidenčna matrika je simetrična po diagonali, pri čemer so primeri, kjer sta označevalca primere označila z enakima oznakama, vneseni po diagonali. Na robovih koincidenčne matrike imamo vsote vrednosti po vrstici ali stolpcu, n pa predstavlja število vseh parov oznak.

Tabela 4.4: Primer koincidenčne matrike za dani problem

	K	K	k	
C	o_{ck}	…	…	n_c
C	…	…	…	n_c
C	…	…	…	n_c
	n_k	n_k	n_k	n=2N

Alfa doseže vrednost 1, ko se označevalci popolnoma strinjajo, in vrednost 0, ko se označevalci strinjajo zgolj po naključju.

5 REZULTATI

V tem poglavju predstavljamo rezultate vseh petih naučenih modelov. Kot osnovo za primerjanje zmogljivosti modelov, kot smo jih ocenili z metrikami, opisanimi v poglavju 4.4, smo uporabili modela iz Bučar, Žnidaršič in Povh, 2018. Tudi ta modela sta naučena za prepoznavanje sentimenta v novicah na isti podatkovni zbirki, pri čemer model na podlagi metode podpornih vektorjev (SVM) dosega višjo klasifikacijsko točnost, model, naučen na podlagi metode naivnega Bayesa (NBM), pa dosega višjo oceno F1.

Ker avtorji v izvirnem članku niso posredovali podatkov o točni razdelitvi podatkovne zbirke na učne in testne množice, smo domnevali, da neposredna primerjava modelov iz obeh študij ni povsem realna. Različna razdelitev podatkov na učno in testno množico namreč lahko vpliva na rezultate validacije. Zato smo modela na podlagi metod podpornih vektorjev in naivnega Bayesa, kot sta opisana v izvirnem članku, znova naučili na naši razdelitvi podatkovne zbirke. Na naši razdelitvi sta oba modela dosegla nižje rezultate. Razlog je morda manjša velikost učne množice, ki pri našem eksperimentu predstavlja 80 % izvirnega korpusa, medtem ko so učne množice iz izvirnega članka vsebovale 90 % izvirnega korpusa. Rezultate modelov nevronskih mrež in modelov, ki smo jih vzeli za osnovo, predstavljamo v Tabeli 5.1.

Ker si vsi modeli nevronskih mrež delijo isto arhitekturo, ki združuje dve vrsti značilk, smo se odločili, da izvedemo tudi krajšo študijo, ki primerja, v kakšni meri posamezna vrsta značilk prispeva k rezultatom klasifikacije. Opis in rezultate te študije objavljamo koncu poglavja.

5.1 Rezultati glavne študije

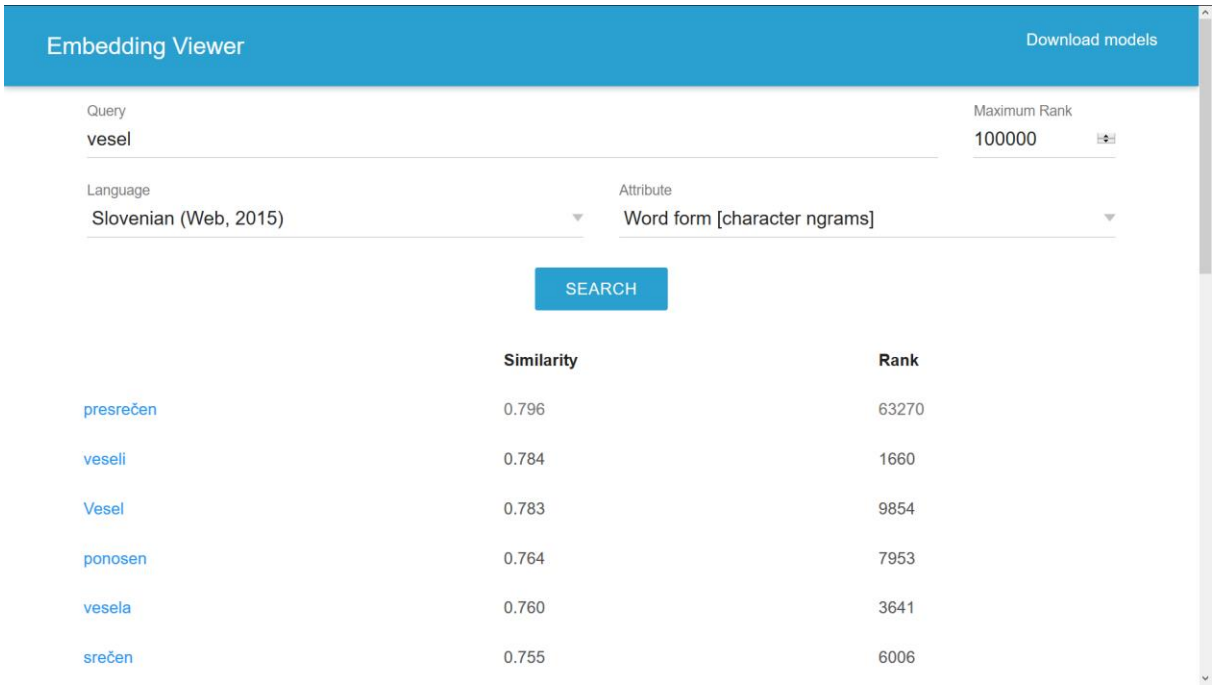
Rezultati modelov, skupaj z rezultati modelov iz Bučar, Žnidaršič in Povh (2018) ter rezultati osnovnih modelov iz ponovljenega poizkusa, so prikazani v Tabeli 5.1. V primerjavi z osnovnimi modeli, ki so bili naučeni na naši razdelitvi podatkovne zbirke, dosegajo vsi modeli nevronskih mrež opazno boljše rezultate tako po klasifikacijski točnosti kot po oceni F1. V primerjavi z modeli iz izvirnega članka je zmogljivost naučenih modelov nevronskih mrež primerljiva s poročanimi rezultati. Modela, ki najslabše klasificirata, LSTM_BOW+TPROC in LSTM_BOW+TPROC+REF, se glede na točnost povsem približata SVM-ju, medtem ko po oceni F1 zaostajata za približno 2 %. Dober rezultat doseže najenostavnejši model LSTM_BOW, ki je po natančnosti izenačen s SVM-jem, čeprav ocene F1 ne izboljša v primerjavi z našima najšibkejšima modeloma. Najboljše rezultate dosegata nevronska modela s popravljenimi vektorskimi vložitvami LSTM_BOW+REF in LSTM_BOW+REF2. Glede na klasifikacijsko točnost sta oba modela povsem izenačena s SVM-jem, pri čemer ga drugi model celo preseže za 0,25 %, po oceni F1 pa se NBM-ju približata na 2,5 % razlike.

Zanimiva je tudi primerjava modelov nevronskih mrež med sabo. Kljub temu, da se metode predobdelave besedila smatrajo kot zelo učinkovite pri učenju klasifikacijskih modelov, se je v naši raziskavi izkazalo ravno nasprotno. Ne samo, da pri modelih LSTM_BOW+TPROC in LSTM_BOW+TPROC+REF, ki uporabljata bolj izrazito predobdelavo vhodnih besedil, ni opaziti izboljšav, zdi se tudi, da sama predobdelava besedila rezultate celo nekoliko poslabša. Vnašanje informacij o sentimentu v vektorske vložitve je imelo nekoliko boljši vpliv. Oba modela, ki sta uporabljala popravljene vektorske vložitve, LSTM_BOW+REF in LSTM_BOW+REF2, sta bila po klasifikacijski točnosti primerljiva z osnovnim SVM-jem, v primerjavi z ostalimi modeli nevronskih mrež pa sta izboljšala oceno F1 za približno 1 %.

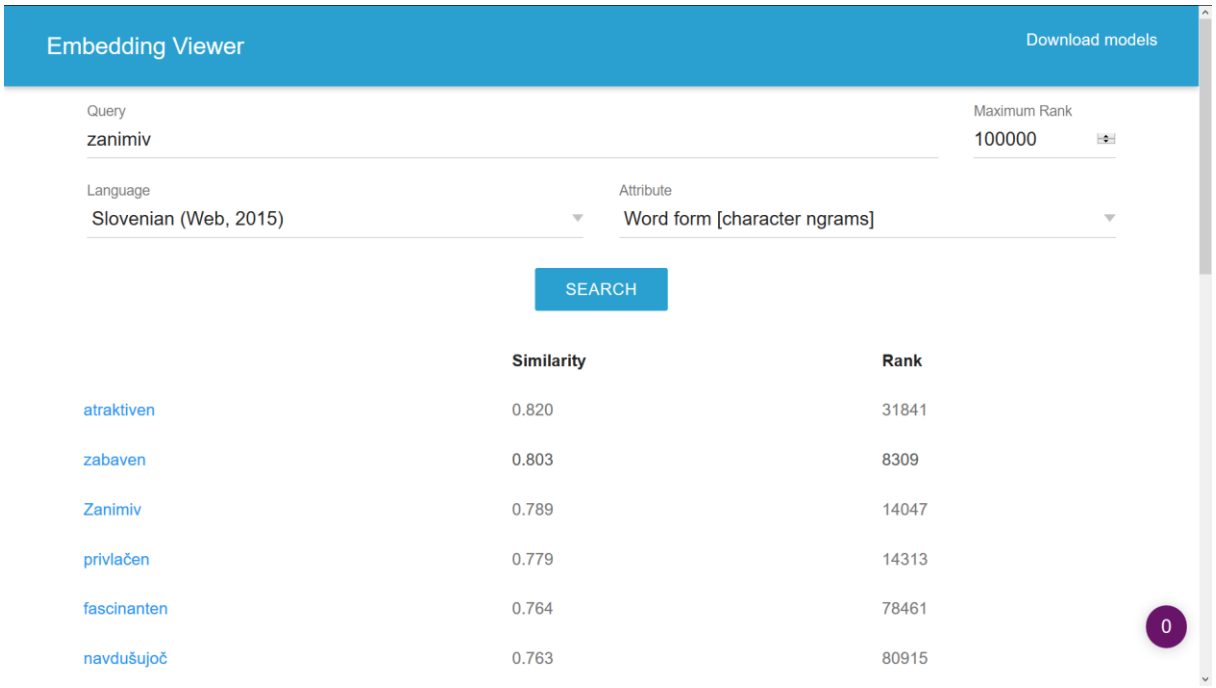
Tabela 5.1: Primerjava rezultatov modelov, naučenih v sklopu te diplomske naloge, in osnovnih modelov s poudarjenimi najboljšimi rezultati, ki smo jih dosegli v okviru trenutne raziskave

Model	**Točnost**	**F1**
SVM (poročano)	66,5 %	63,4 %
NBM (poročano)	64,3 %	65,9 %
SVM (ponovljen poizkus)	62,7 %	59,0 %
NBM (ponovljen poizkus)	53,8 %	24, 57 %
LSTM+BOW	66,5 %	61,6 %
LSTM+BOW+TPROC	66 %	61,1 %
LSTM+BOW+TPROC+REF	65,9 %	61 %
LSTM+BOW+REF	66,4 %	62,1 %
LSTM+BOW+REF2	**66,7 %**	**62,5 %**

Kljub boljšim rezultatom pa je vpliv metode popravljanja vektorskih vložitev glede na sentiment še vedno dokaj zanemarljiv. Predvidevamo, da celotna metoda temelji na preveč optimistični predpostavki. Pri trenutnih kontekstualnih modelih vektorskih vložitev se sicer resda lahko zgodi, da imata dve besedi iste besedne vrste, ki se pojavljata v izjemno podobnih kontekstih, kot na primer besedi *dober* in *slab*, lahko podobno vektorsko vložitev, čeprav izražata nasprotni pomen. Vendar če na hitro poiščemo nekaj takih primerov (glej Sliko 5.1 in Sliko 5.2) in njihovih najbližjih sosedov, hitro opazimo, da je ta primer prej izjema kot pravilo. V bližnji okolici večine besed se namreč nahajajo njihove sopomenke, ki imajo podoben pomen in praktično vedno isti sentiment.



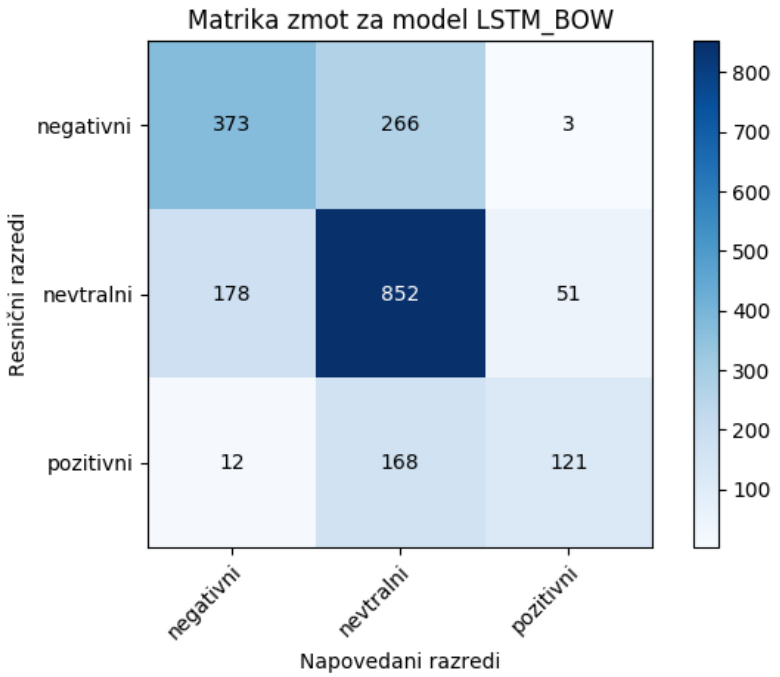
Slika 5.1: Prikaz najbolj podobnih vektorskih vložitev za besedo vesel



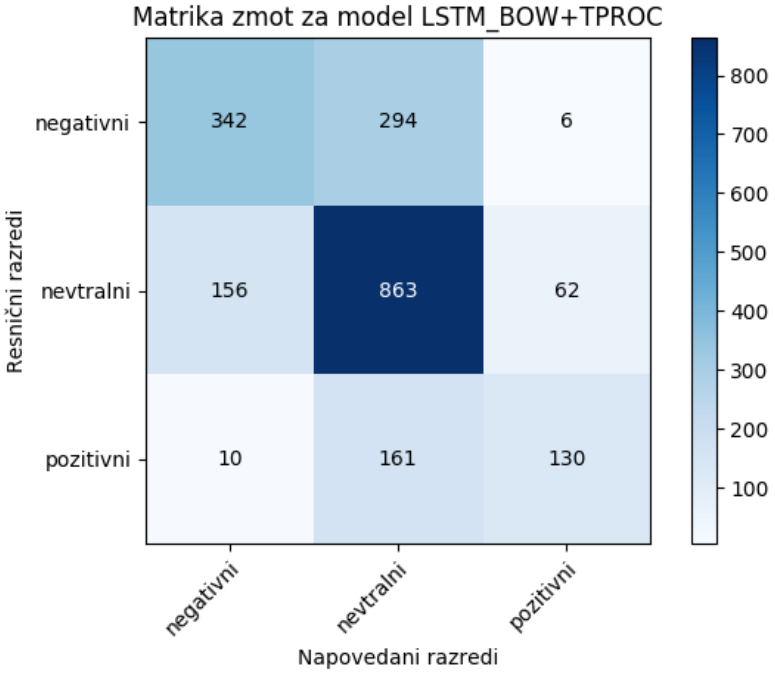
Slika 5.2: Prikaz najbolj podobnih vektorskih vložitev za besedo zanimiv

S pregledom matrik zmot posameznih naučenih modelov (glej Slike 5.3–5.7) pridobimo boljši vpogled v napake, ki jih delajo modeli pri napovedovanju sentimenta. Opazimo lahko, da matrike zmot vseh modelov, ne glede na razlike v predobdelavi podatkov, kažejo dokaj konsistentno sliko. Modeli najmanj napak delajo pri klasifikaciji nevtralnih primerov, nekoliko

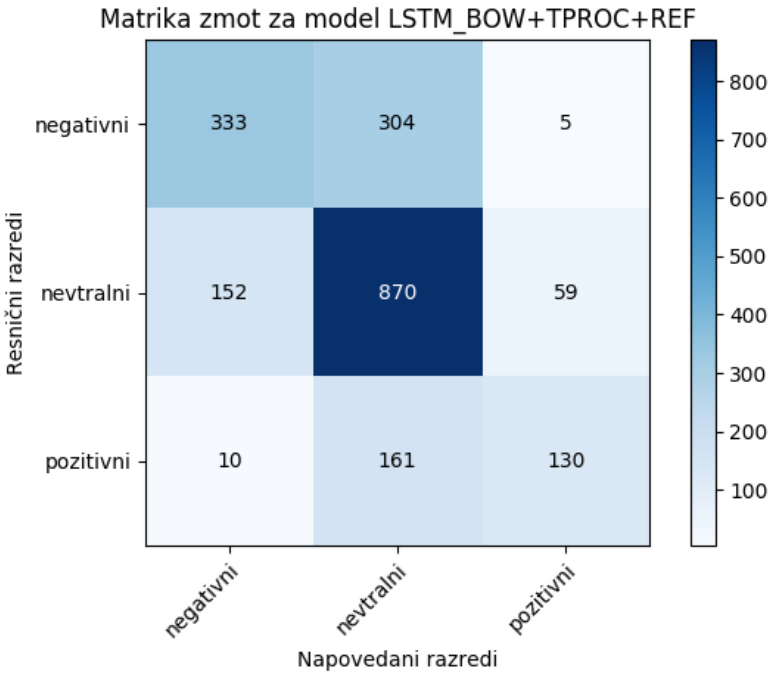
slabše se odrežejo pri klasifikaciji negativnih primerov, najslabše pa klasificirajo pozitivne primere. Ta razporeditev je konsistentna z neuravnoteženostjo primerov v uporabljenem korpusu podatkov, ki v primerjavi z nevtralnimi primeri vsebuje skoraj dvakrat manj negativnih primerov in kar trikrat manj pozitivnih primerov. Nadalje modeli ne delajo veliko napak med dvema skrajnima sentimentoma, negativnim in pozitivnim. Pri vseh modelih je skupno število takih napak na testni množici namreč manjše od 20. Večino napak tako modeli naredijo pri klasifikaciji negativnih in pozitivnih primerov v nevtralne. To je še posebej očitno pri pozitivnem razredu, ki je tudi edini razred, pri katerem vsi modeli več primerov klasificirajo napačno kot pravilno.



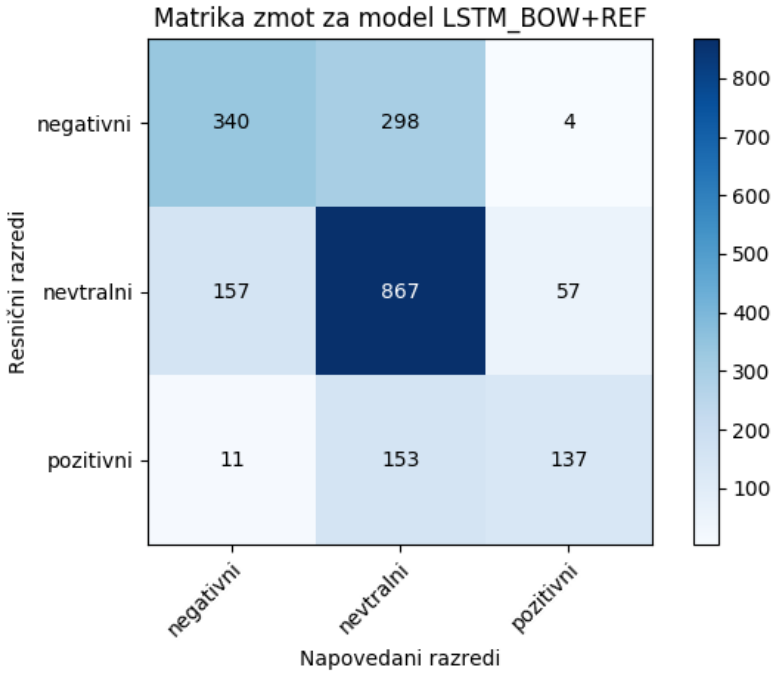
Slika 5.3: Matrika zmot za model LSTM_BOW



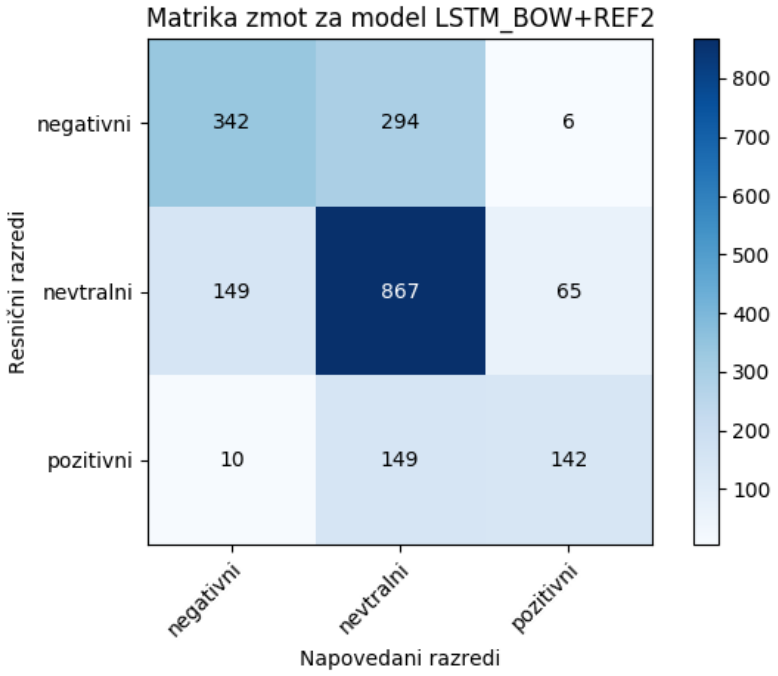
Slika 5.4: Matrika zmot za model LSTM_BOW+TPROC



Slika 5.5: Matrika zmot za model LSTM_BOW+TPROC+REF



Slika 5.6: Matrika zmot za model LSTM_BOW+REF



Slika 5.7: Matrika zmot za model LSTM_BOW+REF2

Rezultate zadnje mere, Krippendorfove alfe, predstavljamo posebej v Tabeli 5.2, ker je ne moremo neposredno primerjati z osnovnimi modeli, saj v izvirnem članku za te modele ni poročana. Naši modeli dosegajo podobne vrednosti v obsegu 0,52 do 0,54, kar označuje zmerno

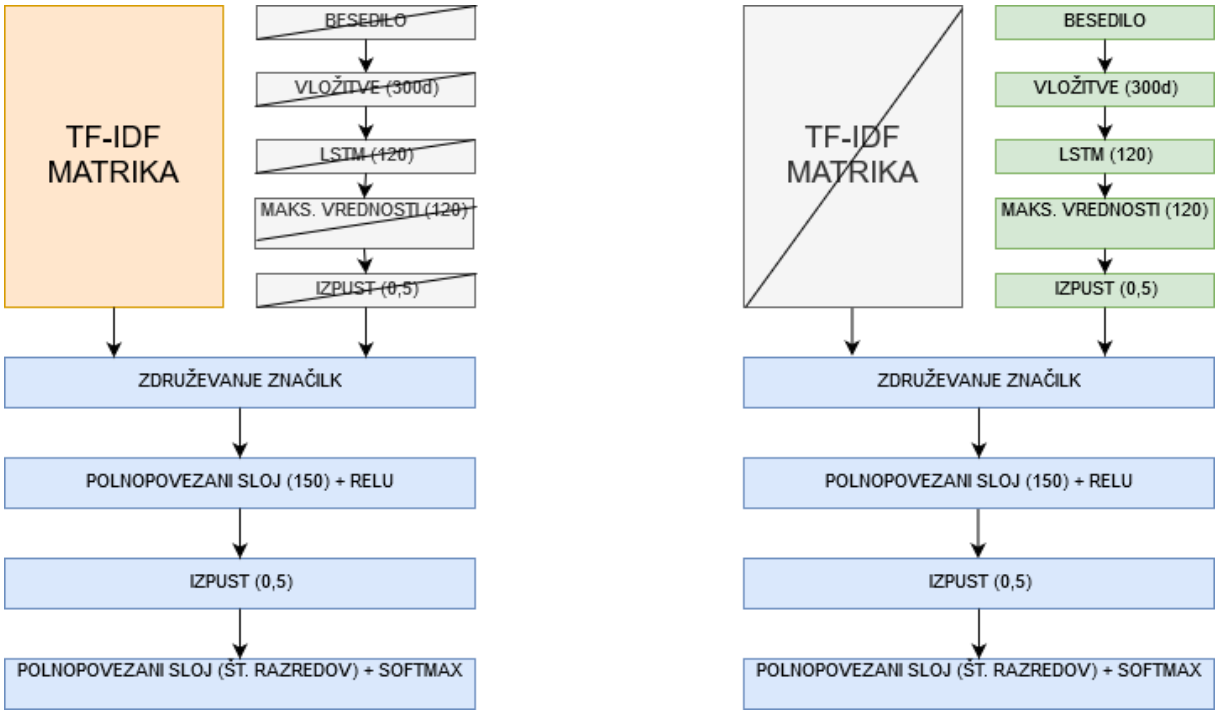
strinjanje z oznakami iz zlatega standarda. Čeprav matrika zmot kaže na precej malo napak med dvema skrajnima razredoma, ki bi bolj vplivale na nižjo vrednost Krippendorffove alfe, ta še vedno ostaja zmerna, kar kaže na precej veliko negotovost modelov pri klasifikaciji v nevtralni razred. Tudi pri tej meri se znova pokažejo trendi kot pri drugih merah, in sicer najmanjše vrednosti, 0,52 dosegajo modeli s temeljitejšo predobdelavo besedil, največjo, 0,54 pa model s popravljenimi vektorskimi vložitvami.

Tabela 5.2: Krippendorfova alfa modelov nevronskih mrež, naučenih v okviru trenutne raziskave s poudarjeno najvišjo doseženo vrednostjo

Model	**Krippendorfova α**
LSTM+BOW	0,53
LSTM+BOW+TPROC	0,52
LSTM+BOW+TPROC+REF	0,52
LSTM+BOW+REF	0,53
LSTM+BOW+REF2	**0,54**

5.2 Rezultati primerjave napovedne učinkovitosti posamezne skupine značilk

V zadnjem delu tega poglavja bomo predstavili še rezultate primerjave napovedne učinkovitosti posamezne skupine značilk nevronskih modelov. Vsi naučeni modeli si delijo isto arhitekturo z dvema obdelovalnima cevovodoma. En cevovod je sestavljen iz rekurentnih nevronskih mrež z dolgoročnimi in kratkoročnimi pomnilnimi celicami, ki iz vhodnih besedil samodejno izluščijo značilke, drugi cevovod pa vhodna besedila predstavi v obliki TF-IDF matrike. Napovedno učinkovitost posamezne vrste značilk smo preverjali tako, da smo iz arhitekture odstranili en obdelovalni cevovod in model na učni množici naučili samo s pomočjo preostalih značilk. Za vsak model smo nato izračunali klasifikacijsko točnost in oceno F1 ter rezultate primerjali. Shemo eksperimenta predstavlja Slika 5.8.



Slika 5.8: Priprava arhitekture modelov za primerjavo vpliva značilk na klasifikacijo

Kljub temu, da imamo samo dva cevovoda, smo v tej študiji naučili tri modele. En model je vseboval samo cevovod z rekurentnimi nevronskimi mrežami, dva pa s TF-IDF utežmi. Razlika med modeloma s TF-IDF utežmi je ta, da je bila predobdelava vhodnih besedil pri enem od modelov, po vzoru glavne študije, bolj temeljita. Rezultati vseh treh modelov z opisom predobdelave besedil so predstavljeni v Tabeli 5.3.

Tabela 5.3: Rezultati primerjave napovedne učinkovitosti posamezne vrste značilk

Značilke	**Predobdelava besedila**	**Točnost**	**F1**
Samodejno generirane z LSTM plastmi	Sprememba velikih črk v male	62,5 %	62,5 %
TF-IDF matrika	Sprememba velikih črk v male	**65 %**	**65 %**
TF-IDF matrika	Sprememba velikih črk v male, odstranitev številk, neinformativnih besed in ločil	64,7 %	64,7 %

Rezultati kažejo, da so TF-IDF uteži za podani problem nekoliko bolj učinkovite, saj so rezultati za približno 2,5 % boljši, kot če značilke generira rekurentna nevronska mreža. Kljub temu se kombiniranje obeh vrst značilk še vedno izkaže za najučinkovitejše, saj noben od modelov, ki uporablja samo eno vrsto značilk, ne dosega učinkovitosti modelov iz glavne študije.

Če primerjamo modela s samo TF-IDF utežmi, opazimo, da je pri modelu, ki kot vhod prejme temeljiteje predobdelano besedilo, opazen rahel padec učinkovitosti v primerjavi z drugim modelom. Čeprav je odstopanje izjemno majhno, rezultat tudi tukaj nakazuje, da predobdelava besedil na tem problemu poslabša napovedno učinkovitost končnega modela, kar se ne sklada z ugotovitvami iz literature.

6 ZAKLJUČEK

V okviru diplomske naloge smo se osredotočili na problem samodejnega zaznavanja sentimenta v slovenskih novicah, problema, ki zaenkrat še ni bil podrobno raziskan. Cilj naloge je bil razvoj modela, ki bi uspešno napovedal sentiment v novicah, kot ga dojema povprečen slovenski bralec. Problem smo zastavili kot klasifikacijski problem, kjer smo poskušali novice kategorizirati v eno od treh kategorij: negativno, pozitivno in nevtralno. V ta namen smo razvili arhitekturo nevronskih mrež, ki posamezno novico predstavi na dva načina, in sicer s pomočjo vektorskih vložitev in kot matriko n-gramov s TF-IDF utežmi. Kot zlati standard za učenje modela smo vzeli korpus novic SentiNews, ki vsebuje 10.427 novic, ročno označenih s sentimentom. Skupaj smo naučili šest modelov z enako arhitekturo, ki so se rahlo razlikovali v predstavitvi besedil. Kot osnovo za primerjavo rezultatov smo vzeli rezultate modelov iz Bučar, Žnidaršič in Povh, 2016, ki so po naših informacijah tudi edini modeli, ki se ukvarjajo s podobno problematiko. Ker so bili modeli v članku naučeni na drugačni razdelitvi podatkovne zbirke, smo modele, kot so opisani v izvirnem članku, tudi sami naučili na naši razdelitvi. Modele nevronskih mrež smo nato primerjali tako z rezultati iz ponovljenega poskusa kot z rezultati modelov iz izvirnega članka.

Rezultati raziskave so vzpodbudni, saj so naši modeli rezultate iz ponovljene študije gladko presegli. Poleg tega so dosegli povsem primerljivo klasifikacijsko točnost z najboljšimi

osnovnimi modeli iz izvirnega članka ter se jim po oceni F1 povsem približali. Tako smo potrdili prvo hipotezo, da so lahko nevronske mreže konkurenčne tudi na področju analize sentimenta, kjer navadno prevladujejo bolj tradicionalne metode strojnega učenja, predvsem metoda podpornih vektorjev. Hkrati smo potrdili tudi drugo hipotezo, in sicer, da se lahko modeli na osnovi nevronskih mrež, ki navadno zahtevajo večje količine podatkov, uspešno učijo tudi na manjših korpusih. Vse naše modele smo naučili na korpusu, ki vsebuje zgolj okrog 10.000 podatkov.

Za konec smo opravili še pregled vpliva posamezne vrste značilk na rezultate klasifikacije. Študijo smo opravili tako, da smo naučili tri modele, pri vsakem pa izločili eno od vrst značilk. Skladno s preverjeno literaturo so rezultati pokazali, da so n-grami s TF-IDF utežmi nekoliko bolj diskriminativni za klasifikacijo kot samodejno generirane značilke rekurentnih plasti. Kljub temu modeli s samo eno vrsto značilk ne dosegajo tako visoke klasifikacijske točnosti kot modeli, ki uporabljajo kombinacijo obeh.Verjamemo, da arhitektura, uporabljena v tej diplomski nalogi, še ni dosegla mej zmogljivosti. Za začetek bi bilo treba hiperparametre modela dodatno optimizirati. Treba je namreč pripomniti, da smo vrednosti hiperparametrov izbrali glede na vrednosti, ki v praksi dosegajo ugodne rezultate, niso pa nujno optimalne za dani problem. S pomočjo tehnike mrežnega iskanja (ang. *grid search*) ali naključnega iskanja (ang. *random search*) bi lahko poskusili te vrednosti optimizirati, kar bi lahko prineslo nekaj dodatnih odstotkov pri klasifikacijski točnosti.

Študije, navedene v poglavju 2.2, poročajo o ugodnih rezultatih analize sentimenta v novicah ob vnosu dodatne informacije o sentimentu v model. V naši študiji smo to dosegli tako, da smo informacijo o sentimentu vnesli v vektorske vložitve, in sicer tako, da smo jih na podlagi informacij iz leksikona sentimenta prilagodili v vektorskem prostoru. Drugi avtorji poročajo o dobrih rezultatih pri vpeljavi informacij iz leksikona sentimenta neposredno v TF-IDF matriko. Ker naša arhitektura kot vhod prejme tako samodejno generirane značilke kot TF-IDF uteži, bi lahko dodatna vpeljava informacij o sentimentu v TF-IDF matriko ali kot dodatne značilke pozitivno vplivala na rezultate klasifikacije.

Pogosto pristop za zmanjševanje dimenzionalnosti vhodnih podatkov je izbira najbolj reprezentativnih značilk. To omogoča, da dodatno zmanjšamo šum, ki ga vnesemo v model, in načeloma pozitivno vpliva na rezultate klasifikacije. V okviru naše raziskave ta postopek sicer izvajamo na samodejno generiranih značilkah, in sicer tako, da izberemo zgolj značilke z

najvišjimi vrednostmi, ne izvajamo ga pa na matriki s TF-IDF obtežitvami. Tako bi lahko na primer za vsako TF-IDF utež v matriki izračunali korelacijo z napovednimi razredi in modelu podali samo N značilk, ki najbolje korelirajo z napovednim razredom. Pri tem pristopu je treba sicer pripomniti, da na ta način uvajamo še en hiperparameter, in sicer število izbranih značilk, ki bi ga bilo treba dodatno optimizirati.

Cilj diplomske naloge je bil razviti model nevronskih mrež, ki bi se pri analizi sentimenta v novicah čim bolj približal človeški presoji, kar smo tudi dosegli. Naučeni model, ki je dosegel najboljše rezultate, je na voljo kot spletna storitev na spletnem naslovu classify.ijs.si prek dostopne točke /sentiment/slo_sentiment. Poleg tega objavljamo izvorno kodo in naučene uteži za vse modele na spletnem naslovu https://gitlab.com/Andrazp/analiza_sentimenta_v_novicah.git. Koda omogoča ponovitev poizkusa na testni množici ali uporabo naučenih modelov na povsem novi zbirki podatkov.

7 LITERATURA IN VIRI

24ur.com. (2012). *Kako merimo politični indeks kandidatov?* Pridobljeno iz https://www.24ur.com/novice/slovenija/kako-merimo-politicni-indeks-kandidatov.html (27. 8. 2019).

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M, Kudlur, M, Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI}16)*, str. 265–283.

Baccianella, S., Esuli, A. & Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *Lrec*, *2010(10)*, str. 2200–2204.

Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Van Der Goot, E., Halkia, M., Pouliquen, B. & Belyaeva, J. (2013). *Sentiment analysis in the news.* Pridobljeno iz https://arxiv.org/abs/1309.6202 (18. 9. 2019).

Beigi, G., Hu, X., Maciejewski, R. & Liu, H. (2016). An overview of sentiment analysis in social media and its applications in disaster relief. *Sentiment analysis and ontology engineering*, str. 313–340.

Bučar, J., Povh, J. & Žnidaršič, M. (2016). Sentiment classification of the Slovenian news texts. *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, str. 777–787.

Bučar, J., Žnidaršič, M. & Povh, J. (2018). Annotated news corpora and a lexicon for sentiment analysis in Slovene. *Language Resources and Evaluation*, *52(3)*, str. 895–919.

CS231n Convolutional Neural Network for Visual Recognition. Pridobljeno iz http://cs231n.github.io/ (9. 9. 2019).

Cortis, K., Freitas, A., Daudert, T., Huerlimann, M., Zarrouk, M., Handschuh, S. & Davis, B. (2017). Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, str. 519–535.

Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding.* Pridobljeno iz https://arxiv.org/abs/1810.04805 (18. 9. 2019).

Grave, E., Bojanowski, P., Gupta, P., Joulin, A. & Mikolov, T. (2018). *Learning word vectors for 157 languages.* Pridobljeno iz https://arxiv.org/abs/1802.06893 (18. 9. 2019).

Hu, M. & Liu, B. (2004). Mining and Summarizing Customer Reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, str. 168–177.

Kaur, G. & Kaur, K. (2015). Sentiment Analysis on Punjabi News Articles Using SVM. *Int. J. Sci. Res., 6(8)*, str. 414–421.

Kingma, D. P. & Ba, J. (2014). *Adam: A method for stochastic optimization.* Pridobljeno iz https://arxiv.org/abs/1412.6980 (18. 9. 2019).

Krippendorff, K. (2011). Computing Krippendorff 's Alpha-Reliability. Pridobljeno iz http://repository.upenn.edu/asc_papers/43 (28. 8. 2019).

Kumar, A., Sethi, A., Akhtar, M. S., Ekbal, A., Biemann, C. & Bhattacharyya, P. (2017, avgust). IITPB at SemEval-2017 task 5: Sentiment prediction in financial text. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, str. 894–898.

Li, X., Xie, H., Chen, L., Wang, J. & Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems, 69*, str. 14–23.

Lin, K. H. Y., Yang, C. & Chen, H. H. (2008). Emotion classification of online news articles from the reader's perspective. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, str. 220–226.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies, 5(1)*, str. 1–167. Pridobljeno iz https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf (27. 8. 2019).

Manning, C., Raghavan, P. & Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering, 16(1)*, str. 100–103.

Mansar, Y., Gatti, L., Ferradans, S., Guerini, M. & Staiano, J. (2017). *Fortia-FBK at SemEval-2017 task 5: Bullish or bearish? Inferring sentiment towards brands from financial news headlines.* Pridobljeno iz https://arxiv.org/abs/1704.00939 (18. 9. 2019).

Martinc, M. & Pollak, S. (2019). Combining n-grams and deep convolutional features for language variety classification. *Natural Language Engineering*, str. 1–26.

Mejova, Y. (2009). *Sentiment analysis: An overview*. University of Iowa, Computer Science Department.

Moore, A. & Rayson, P. (2017). *Lancaster A at SemEval-2017 Task 5: Evaluation metrics matter: predicting sentiment from financial news headlines.* Pridobljeno iz https://arxiv.org/abs/1705.00571 (18. 9. 2019).

Mozetič, I., Grčar, M. & Smailović, J. (2016). Multilingual Twitter sentiment classification: The role of human annotators. *PloS one, 11(5)*, e0155036.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press. Pridobljeno iz http://neuralnetworksanddeeplearning.com/index.html (9. 9. 2019).

Novak, P. K., Smailović, J., Sluban, B. & Mozetič, I. (2015). Sentiment of emojis. *PloS one, 10(12)*, e0144296.

Pelicon, A., Martinc, M. & Novak, P. K. (2019). Embeddia at SemEval-2019 Task 6: Detecting Hate with Neural Network and Transfer Learning Approaches. *Proceedings of the 13th International Workshop on Semantic Evaluation*, str. 604–610.

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A comprehensive grammar of the English language*. Harlow: Longman.

Ruder, S. (2016). *An overview of gradient descent optimization algorithms.* Pridobljeno iz https://arxiv.org/abs/1609.04747 (18. 9. 2019).

Rutar, S. (2016). Empirična evalvacija procesa avtomatske klasifikacije sentimenta na finančni domeni (diplomska naloga). Pridobljeno iz https://repozitorij.uni-lj.si/Dokument.php?id=95384&lang=slv (27. 8. 2019)

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 conference on empirical methods in natural language processing*, str. 1631–1642.

Taj, S., Shaikh, B. B. & Meghji, A. F. (2019). Sentiment Analysis of News Articles: A Lexicon based Approach. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, str. 1–5.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T. & Zhou, M. (2015). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering, 28(2)*, str. 496–509.

Tripathy, A., Agrawal, A. & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, str. 117–126.

Colah's blog. (2015). *Understanding LSTM networks*. Pridobljeno iz https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (9. 9. 2019).

Uysal, A. K. & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50(1)*, str. 104–112.

Van de Kauter, M., Breesch, D. & Hoste, V. (2015). Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Systems with applications*, *42(11)*, str. 4999–5010.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. & Hovy, E. (2016). Hierarchical attention networks for document classification. *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, str. 1480–1489.

Yu, L. C., Wang, J., Lai, K. R. & Zhang, X. (2017). Refining word embeddings for sentiment analysis. *Proceedings of the 2017 conference on empirical methods in natural language processing*, str. 534–539.