Duboko učenje za ekstraktivno sažimanje teksta

Aljević, Dino

Sveučilište u Rijeci - Odjel za informatiku

Informacijski i komunikacijski sustavi

Dino Aljević

Duboko učenje za ekstraktivno sažimanje teksta

Diplomski rad

Mentor: prof. dr. sc. Sanda Martinčić-Ipšić

Rijeka, listopad 2021.

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 28.4.2021.

Zadatak za diplomski rad

Pristupnica: Dino Aljević

Naziv diplomskog rada Duboko učenje za ekstraktivno sažimanje teksta

Naziv diplomskog rada na eng. jeziku Deep Learning for Extractive Text Summarization

Sažetak teme / sadržaj zadatka diplomskog rada

Zadatak diplomskog rada je postaviti postupke ekstraktivnog sažimanja tekstualnih dokumenata, gdje se sažeci dobivaju izlučivanjem i rangiranjem najistaknutijih rečenica izvornog dokumenta. Cilj rada je usporediti performanse jednostavne feed-forward neuronske mreže, odnosno višeslojnog softmax klasifikatora koji određuje sažetak na temelju ulaznih Doc2Vec vektora s usporedivom dubokom neuronskom mrežom. Performanse naučenih mreža će se usporediti međusobno te s osnovnim nenadziranim metodama poput sažimanja temeljem reprezentacije u grafovima rečenica koje određuju sažetak na temelju selektivnosti čvorova grafa. U grafovima rečenica se bridovi uspostavljaju temeljem mjera sličnosti (Jaccard, Mihalcea i kosinusna sličnost) između rečenica. Postupci će se razviti i testirati na skupu podataka BBC News Summary ili sličnom odgovarajućem skupu podataka primjerenim za evaluaciju sažimanja teksta.

Standardni pristup mjerenja performansi tehnika automatskog sažimanja teksta temelji se na ROUGE. To je princip vrednovanja usmjeren na opozivu, zasnovan na dobro poznatoj mjeri strojnog prijevoda BLEU-a koji kvantificira preklapanje n-grama u sažetku zlatnog standarda (tj. ručno pripremljenog sažetka) s n-gramima u automatski generiranom sažetku. Odabrane mjere su ROUGE-1, ROUGE-2 i ROUGE-LCS.

Mentorica:
prof. dr. sc. Sanda Martinčić-Ipšić

Voditeljica za diplomske radove:
Izv. prof. dr. sc. Ana Meštrović

Zadatak preuzet: 1.7.2021.

(potpis pristupnika)

Sažetak

Ekstraktivna sumarizacija teksta je proces automatske izgradnje sažetka izlučivanjem najistaknutijih rečenica iz izvornog teksta. U ovom radu su opisane dvije metode za ekstraktivnu sumarizaciju teksta bazirane na dubokom učenju: višeslojni perceptron i upravljačka rekurentna jedinica. Obje metode klasificiraju ulazne rečenične vektore u binarnu klasu ovisno pripada li rečenica sažetku ili ne. Kod perceptrona svaka rečenica u tekstu je predstavljena Doc2Vec vektorom. Metoda upravljačke rekurentne jedinice uči vrijednosti parametara prikrivenih stanja iz vektora ulaznih nizova riječi. Embeding sloj se inicijalizira s vrijednostima Word2Vec vektora svih riječi u vokabularu, a na izlaznom sloju se rečenice klasificiraju u sažetak. U postupcima učenja korišten je standardni skup podataka CNN/DailyMail, a sumarizacija je evaluirana standardnom mjerom za procjenu kvalitete generiranog sažetka - ROUGE. Generalno, višeslojni perceptron postiže bolje rezultate ukoliko se uvažava odziv, dok GRU postiže preciznije rezultate bez obzira na ROUGE mjeru evaluacije. Iako daleko od mogućnosti tvorenja optimalnih sažetka, rezultati ukazuju na uspješnost ovih metoda za problem ekstraktivne sumarizacije teksta.

Ključne riječi: sažimanje teksta, neuronska mreža, duboko učenje, višeslojni perceptron, rekurentna neuronska mreža, upravljačka rekurentna jedinica.

Abstract

Extractive text summarization is tasked with the automatic creation of a summary by extracting the most salient sentences from the original text. In this thesis, two extractive summarization methods are trained and tested: multilayer perceptron and gated recurrent unit. Both methods are trained as a binary classifier capable of assigning the class summary or not-summary to input sentences. In the multilayer perceptron method, each sentence is represented by Doc2Vec embedding. The embedding layer of GRU is initialized with Word2Vec vectors of the words in the vocabulary and the input vectors contain sequences of the words from the original text. The output is a class probability assigned by the logistic function classifier. The CNN/DailyMail dataset is used to train and evaluate extractive summarization models using the ROUGE-1, ROUGE-2, and ROUGE-LCS measures to assess the performance. Generally, GRU achieves better extractive summarization results when precision is considered, while perceptron performs better according to the recall metrics, regardless of the used ROUGE measures. The results are indicating that both methods are capable of performing extractive summarization task.

Keywords: text summarization, neural network, deep learning, multilayer perceptron, recurrent neural network, gated recurrent unit.

Popis korištenih kratica

BOW	Bag-of-words	Neuređena vreća riječi
CBOW	Continious bag-of-words	Neprekidna vreća riječi
GRU	Gated recurrent unit	Upravljačka rekurentna jednica ili propusna povratna ćelija
LSTM	Long short-term memory	Ćelija s dugoročnom memorijom
MLP	Multi layer perceptron	Višeslojni perceptron
NNLM	Neural network language model	Neuronski jezični model
	Embedding	Gusta reprezentacija
	Deep feed-forward network	Duboka unaprijedna mreža
PV	Paragraph vector	Vektor odlomka
PV-DBOW	Paragraph vector - distributed bag-of-words	Distribuirani vektor odlomka
PV-DM	Paragraph vector - distributed memory	Memorijski vektor odlomka
RNN	Recurrent neural network	Povratne neuronske mreže ili Rekurentne neuronske mreže
ROUGE	Recall-oriented understudy for gisting evaluation	

Sadržaj

Popis slika

Popis tablica

1 Uvod

Živimo u vrijeme eksponencijalnog rasta količine podataka – u vrijeme podataka velikog opsega (*big data*) – u kojima značajan udio imaju nestrukturirani podaci, dakle tekstovi, koje je potrebno prilagoditi za brza i učinkovita pretraživanja, indeksiranja, grupiranja pa i brzo čitanje sažetih informacija iz teksta. Metode izlučivanja ključnih riječi [1], [2] i sažimanja teksta [3], [4] postaju od iznimne važnosti za računalne postupke obrade velikih količina tekstova u različitim domenama poput poslovnih dopisa, e-pošte, medicinske dokumentacije, mišljenja korisnika na društvenim mrežama i sl.

Postupci sažimanja teksta tradicionalno koriste postupke strojnog učenja [5] i/ili reprezentacije teksta u grafu [6]. U novije vrijeme je značajan napredak postignut korištenjem postupaka dubokog strojnog učenja (*deep learning*) odnosno različitim arhitekturama dubokih neuralnih mreža (*deep neural networks*) [3], [7]. Osim uvođenjem različitih arhitekta dubokih mreža napredak je postignut i uvođenjem različitih načina reprezentacije tekstova u formi niskodimenzionalnih vektora u neprekidnom prostoru - gustim reprezentacijama odnosno embedingsima (*embeddings*) [8], [9].

Općenito razlikujemo ekstraktivnu i apstraktivnu sumarizaciju [10]. Ekstraktivna sumarizacija otkriva najvažnije rečenice ili dijelove rečenica koji su već zapisani u tekstu. Dok je apstraktivnoj cilj iz automatski izvučenih informacija ili dijelova teksta tvoriti novi tekst sažetka. Nadalje sumarizacija se može vršiti iz jednog ili iz kolekcije dokumenata, pri čemu dokumenti mogu biti pisani i na više različitih jezika (npr. sažeci znanstvenih radova su pored izvornog jezika često dostupni i na engleskome jeziku).

Cilj diplomskog rada je postaviti postupke ekstraktivnog sažimanja tekstova dokumenata. U radu se uspoređuju performanse višeslojnog perceptrona, odnosno binarnog klasifikatora koji određuje sažetak na temelju ulaznih Doc2Vec vektora s usporedivom dubokom neuronskom mrežom u RNN (*Recurrent neural network*) arhitekturi, odnosno u njenoj poboljšanoj GRU (*Gated recurrent unit*) varijanti, dakle u arhitekturi s upravljačkom rekurentnom jedinicom. Performanse naučenih modela evaluiraju se standardnim ROUGE postupcima evaluacije sažimanja teksta na CNN/DailyMail skupu podatka.

Diplomski rad opisuje višeslojni perceptron u poglavlju 2 te arhitekturu rekurentne neuronske mreže i upravljačke rekurentne jedinice u poglavlju 3. Postupci učenja gustih reprezentacija Word2Vec i Doc2Vec opisani su u poglavlju 4. Korišteni skup podataka i njegova statistika je prikazana u poglavlju 5. Poglavlje 6 opisuje postupke evaluacije, dok su korišteni alati i implementacijski detalji navedeni u poglavlju 7. Rezultati ekstraktivnog sažimanja su navedeni i objašnjeni u poglavlju 8. Diplomski rad završava zaključkom i smjernicama za daljnji rad u poglavlju 9.

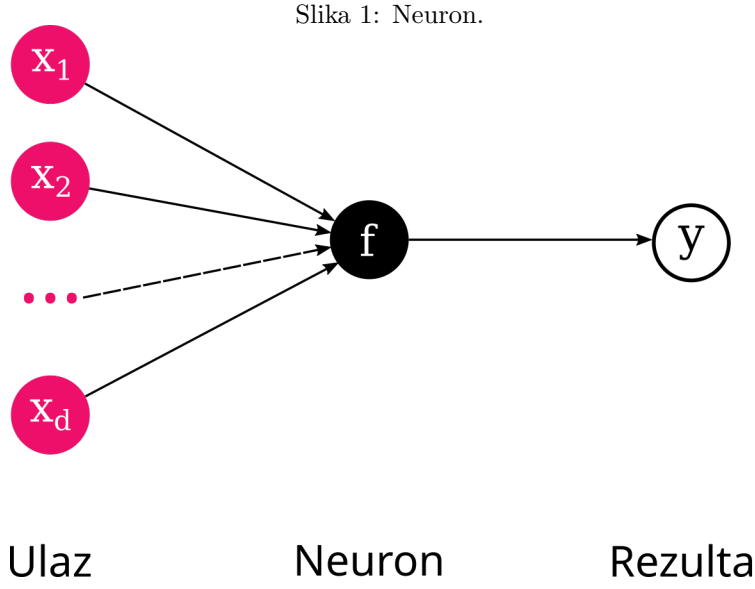
2 Perceptron

Perceptron je naziv za najjednostavniju vrstu neuronske mreže koja je linearni model [7]:

$$NN_{perceptron}(x) = xW + b \tag{1}$$

gdje je $x \in \mathbb{R}^d$ ulazni vektor, $W \in \mathbb{R}^{d_{in} \times d_{out}}$ matrica težina, a $b \in \mathbb{R}^{d_{out}}$ vektor pristranosti (*bias*). Kao što samo ime govori, neuronske mreže su dobile naziv po neuronima, tj. skupovima povezanih ćelija u ljudskom mozgu koje prenose električni signal i ključne su u funkciji mozga. Svedeno na matematički model, svaki neuron je jedinica koja može izvoditi računske operacije, odnosno ona množi ulazne vrijednosti x s težinama w, a zatim suma tih produkata "prolazi" kroz nelinearnu funkciju (nije prisutna u jednadžbi 1). Kako bi stvorili mrežu, neuroni se povezuju tako da izlaz iz jednog neurona postaje ulaz u drugi neuron. Na ovaj način mogu se aproksimirati i složenije funkcije [7].

Slika 1: Neuron.



Perceptroni se tipično sastoje od većeg broja neurona, no oni se uvijek povezuju tako da je tok informacija, odnosno podataka u istom smjeru. Generalno govoreći, neuralne mreže čiji tok ide u jednom smjeru se zovu unaprijedne (*feed-forward*) mreže.

2.1 Višeslojni perceptron

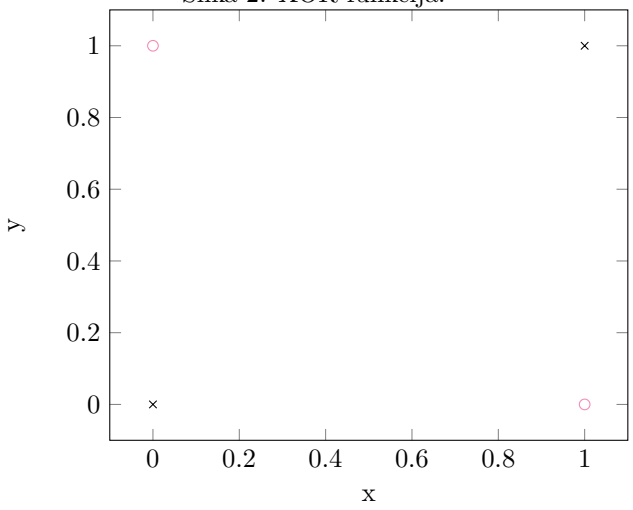
Perceptroni su linearni modeli, te kao takvi imaju problema s modeliranjem nelinearnih problema, npr. perceptron nikako ne može aproksimirati XOR funkciju budući da je ona linearno nerazdvojiva [11]. Neka je $XOR(x, y)$ funkcija defini-

rana nad domenom $\{0, 1\}$:

$$XOR(x, y) = \begin{cases} 0, & \text{ako } x = y \\ 1, & \text{inače} \end{cases} \tag{2}$$

Iz slike 2 trivijalno je zaključiti da ne postoji pravac p tako da razdvaja presliku funkcije u dva disjunktna skupa za svaku permutaciju x i y.

Slika 2: XOR funkcija.



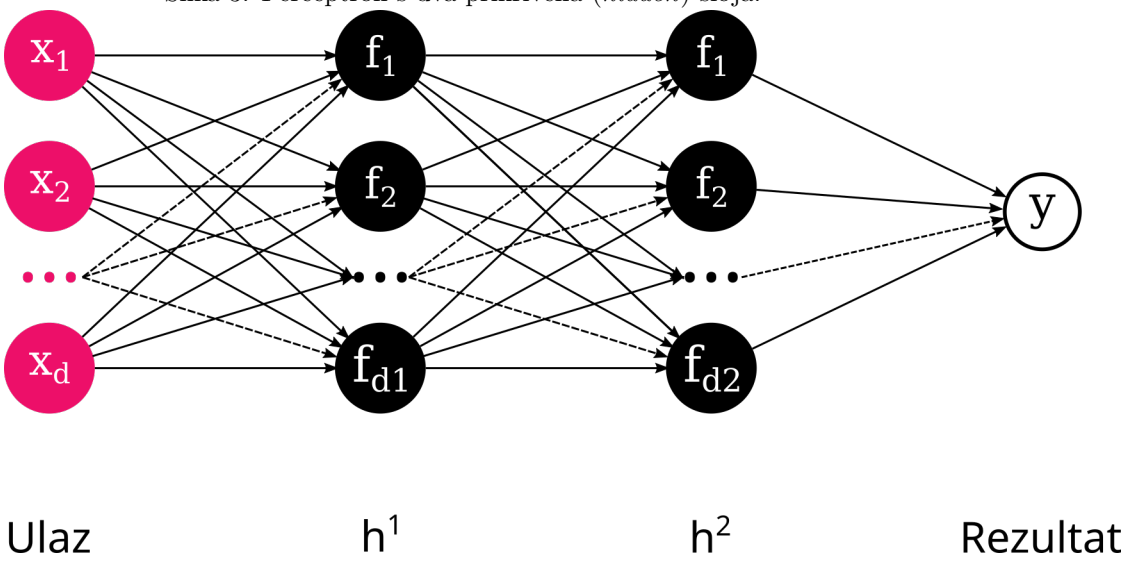
Za rješavanje nelinenarnih problema perceptronu se obično dodaju "slojevi" neurona, npr. dvoslojni perceptron [7]:

$$MLP_1(x) = g(xW^1 + b^1)W^2 + b^2. \tag{3}$$

Eksponenti parametara u jednadžbi se referiraju na redni broj linearne transformacije ulaznih vrijednosti, a g je nelinearna funkcija, tzv. aktivacijska funkcija [7]. Kažemo da ovakav perceptron ima dva sloja, odnosno dva vektora koji su rezultat linearnih transformacija, npr. $xW^1 + b^1$ za prvi sloj, te su oni potpuno povezani [7]. Generalno perceptroni mogu imati veći broj slojeva, a ako su oni unutranji, onda ih zovemo prikrivenim (*hidden*) slojevima. Zadnji sloj je uvijek izlazni sloj [7]. Nizanjem slojeva postižemo transformaciju ulaznog vektora određenih dimenzija u vektor drukčijih dimenzija, konkretno iz jednadžbe 3, prvi sloj radi transformaciju iz d dimenzije u d_1, a zatim zadnji iz d_1 u izlaznu dimenziju d_2 [7].

Slika 3 prikazuje perceptron s tri sloja, dva prikrivena sloja h^1 i h^2 dimenzija d_1 i d_2, te izlazni sloj y dimenzije 1.

Slika 3: Perceptron s dva prikrivena (*hidden*) sloja.



Primjenjujući nelinearnu funkciju poput sigmoid funkcije na izlazni sloj, perceptron može poslužiti kao binarni klasifikator koji na izlazu daje vjerojatnost neke klase uz pretpostavku da su parametri učenja mreže poput optimizacijskog algoritma i funkcije gubitka (*loss function*) korektno odabrani [7].

3 Rekurentna neuronska mreža

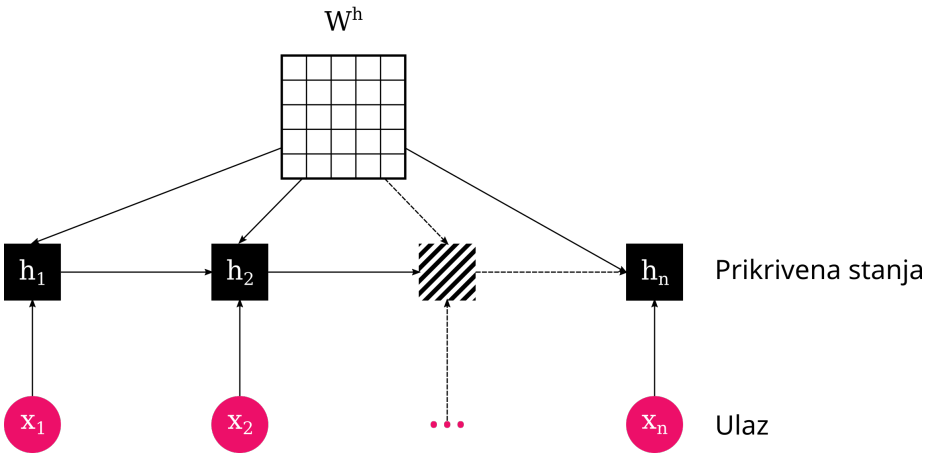
Rekurentne neuronska mreža (*Recurrent neural network, RNN*) je vrsta neuronske mreže s kojom je moguće reprezentirati nizove proizvoljne duljine s vektorima fiksne veličine [7]. Rekurentne neuronske mreže sadrže povratne veze, odnosno omogućavaju da operacije izvedene u prijašnjim koracima utječu na trenutnu operaciju koja se izvodi, a to je korisno kada se obrađuju nizovi poput nizova slova (riječi), nizova riječi (rečenice) itd. Jednostavna rekurentna neuronska mreža je prvi put opisana u [12], te se može matematički može opisati na sljedeći način:

$$h_t = \tanh(W^h h_{t-1} + W^x x_t + b) \tag{4}$$

gdje je $x_t \in \mathbb{R}^{d_x}$ ulazni vektor, $W^x \in \mathbb{R}^{d_x \times d_h}$ matrica težina, $W^h \in \mathbb{R}^{d_h \times d_h}$ matrica stanja, a $b \in \mathbb{R}^{d_h}$ vektor pristranosti. RNN slijedno uči prikrivena stanja h_t za svaki ulazni vektor x_t iz niza $t = 1, 2, ..., n$. Svako prikriveno stanje h_t ovisi o prethodnom stanju h_{t-1}, pa se često stanja nazivaju i vremenskim koracima (*time steps*), odnosno može se reći da se stanje h_t mijena kroz vrijeme. Osim prethodnog stanja h_{t-1} i ulaznog vektora x_t, prikriveno stanje također ovisi i o matrici stanja W^h koja je ista za svaki vremenski korak h_t. Prvo prikriveno stanje h_1 ovisi o "prethodnom" stanju h_0 koje se tipično uzima kao nul-vektor [7]. Na svakom vremensku koraku kao izlaz iz RNN možemo uzeti vektor prikrivenog stanja naučenog u tom koraku, međutim taj vektor samo sadrži naučeno stanje, pa se zato on obično koristi kao ulaz u druge komponente u složenijim arhitekturama dubokih neuronskih mreža [7]. Dakle, RNN se tipično koristi kao jedna komponenta neke složenije arhitekture koja uči parametre poput reprezentacije ulaznih nizova.

Slika 4 prikazuje model jednostavne rekurentne mreže koja uči prikrivena stanje za ulazni niz veličine n.

Slika 4: Model jednostavne rekurentne neuralne mreže.



Očiti problem kod RNN mreža je slijedno učenje težina, tj. algoritam nije moguće paralelizirati budući da svaki novi korak ovisi o onom prethodnom. U praksi to znači da se RNN mreže ne mogu paralelno učiti na distribuiranim sustavima ili grafičkim procesorima, pa njihovo učenje traje dugo čak i za manji broj ulaznih podataka.

3.1 Upravljačka rekurentna jedinica

Najveća mana jednostavne rekurentne neuronske mreže je problem nestajanja gradijenata (*gradient vanishing*) [13] koji otežava učenje ove mreže. Gradijenti izračunati u kasnijim koracima mreže oslabljuju kroz postupak širenja unatrag, pa zbog ovoga jednostavna RNN ne može učiti iz udaljenih ulaznih signala [7]. Zbog problema nestajanja gradijenata jednostavna rekurentna neuronska mreža se ne koristi u praksi, već se ona zamjenjuje s ćelijom s dugoročnom memorijom (*long short-term memory, LSTM*) [14] ili upravljačkom rekurentnom jedinicom (*gated recurrent unit, GRU*) [15] koja se koristi u ovom radu.

Obje arhitekure rješavaju problem nestajanja gradijenata uvođenjem mehanizma pomoću kojeg mreža može kontrolirati utjecaj prošlih stanja (h_{t-1}, h_{t-2}, $..., h_1$) na sadašnje stanje h_t [7]. Na prikriveno stanje je moguće je konceptualno gledati kao na memoriju koja se pročita i preko koje se prepiše vrijednost tijekom svakog koraka [7]. Na početku koraka pročita se memorija (stanje h_{t-1}) zajedno s ulazom x_t, zatim se izvedu računske operacije, te se konačno memorija prepiše s novim stanjem h_t. Ovo znači da se tijekom svakog koraka stari podaci iz memorije prepišu novim, pa se u kontekstu gradijenata ovo može protumačiti kao gubitak informacija o gradijentima. GRU i LSTM uvode koncept propusnica (*gate*), tj. vektora $g \in R^{d_h}$ koji određuje koje dimenzije trenutnog stanja će se ažurirati, a koje će ostati jednake prijašnjem stanju. U GRU arhitekturi jednadžba 4 je proširena na sljedeći način [7]:

$$\begin{aligned} h_t &= u \odot \tilde{h} + (1-u) \odot h_{t-1} \\ u &= \sigma(W^{xu} x_t + W^{hu} h_{t-1}) \\ r &= \sigma(W^{xr} x_t + W^{hr} h_{t-1}) \\ \tilde{h}_t &= \tanh(W^{xh} x_t + W^{hg}(r \odot h_{t-1})) \end{aligned} \tag{5}$$

gdje su $h_{t-1}, h_t \in \mathbb{R}^{d_h}$ stanja u prethodnom, odnosono sadašnjem koraku; $\tilde{h} \in \mathbb{R}^{d_h}$ kandidat za novo stanje; r i u propusnice resetiranja i ažuriranja; a $W^{xu}, W^{xr} \in \mathbb{R}^{d_x \times d_h}, W^{hu}, W^{hr}, W^{hg} \in \mathbb{R}^{d_h \times d_h}$ matrice težina.

r je propusnica resetiranja (*reset gate*) koja se koristi za računanje kandidata novog stanja, vektora $\tilde{h}$ [7]. r je vektor realnih brojeva koji putem Hadamardovog produkta skalira vrijednosti prijašnjeg stanja, te tako određuje komponente koje se propuštaju u novo stanje [7]. Dakle, komponente r čiji iznos je blizu 0 se množe s vektorom h_{t-1}, pa te komponente od prijašnjeg stanja "oslabljuju" u novom stanju, dok one koje se množe s iznosom blizu 1 ostaju približno iste. u je propusnica ažuriranja (*update gate*) koji na sličan način određuje koje komponente kandidata za novo stanje će propustiti, odnosno koje komponente

prijašnjeg stanja će oslabiti [7]. Propusnice su također parametri koje mreža uči, pa njihovi elementi ne mogu biti diskretne vrijednosti (npr. 0 i 1) budući da oni moraju biti diferencijabilni [7]. Ti vektori se "provuku" kroz sigmoid funkciju kako bi poprimili vrijednost između 0 i 1 [7].

4 Reprezentacija teksta

Tekst u svom izvornom obliku ne može poslužiti kao ulaz u neuronsku mrežu budući da se simboli ne mogu koristiti u matematičkim izračunima, stoga je ulaz u neuronsku mrežu numerički vektor koji predstavlja određeni dio teksta. Tradicionalne reprezentacije teksta su bazirane na oskudnim (*sparse*) vektorima visoke dimenzionalnosti gdje svaka dimenzija predstavlja jednu značajku u tekstu, npr. riječ [7]. Popularni model za vektoriziranje teksta je neuređena vreća riječi (*bag-of-words, BOW*) model u kojem histogram riječi, odnosno frekvencija pojavljivanja riječi u tekstu predstavlja značajke, odnosno dimenzije vektora [7]. Ovim modelom obuhvaćamo statističke značajke u tekstu, pri čemu sam poredak riječi ili njihovo značenje ne igra ulogu. Uzmimo primjer teksta koji se sastoji od vokabulara V veličine 5 riječi, tj. $|V| = 5$, te $v_i \in V, i = 1, ..., |V|$. Ako pretpostavimo da se v_1 pojavljuje jedanput, v_2 pet puta, v_3 i v_5 niti jednom, te v_4 dvaput, onda taj tekst možemo predstaviti BOW vektorom $x = \begin{bmatrix}1 & 5 & 0 & 2 & 0\end{bmatrix}$. Korisna BOW reprezentacija individualnih riječi je *one-hot* ili 1-od-V kodiranje gdje su sve dimenzije vektora jednake 0, osim one koja odgovara indeksu te riječi u vokabularu [7]. Recimo da kodiramo riječ $v_3 \in V$, one-hot vektor te riječi izgleda kao $x = \begin{bmatrix}0 & 0 & 1 & 0 & 0\end{bmatrix}$. Ovakve reprezentacije su korisne, ali imaju svoje nedostatke. Za početak, jasno je da dimenzionalnost vektora raste s veličinom vokabulara, a time i vremenska kompleksnost izvođenja računskih operacija. Dakle BOW rezultira s reprezentacijama teksta u obliku visokodimenzionalnih vektora. Nadalje u BOW modelima vjerojatnost da većina dimenzija ima vrijednost 0 raste s veličinom vokabulara, odnosno dijelovi teksta sadrže samo manji dio riječi iz vokabulara, što rezultira problemom oskudnosti podataka (*data sparsity*).

Oko 2014. godine, područje računalne analize prirodnog jezika prolazi kroz važnu promjenu, s kojom je primjena linearnih modela koji rade s oskudnim vektorima visoke dimenzionalnosti zamijenjena nelinearnim modelima koji radi s gustim niskodimenzionalnim vektorima [7]. U tim vektorima, dimenzije više ne predstavljaju značajke teksta, već se značajke ugrađuju u d-dimenzionalni vektorski prostor - *embeddings* [7]. Osim očitih prednosti poput manjeg broja dimenzija i otklonjenog problema oskudnosti, velika prednost ovakvih reprezentacija je njihova moć generalizacije [7], a primjer toga će biti vidljiv u reprezentacijama korištenim u ovom radu, Word2Vec i Doc2Vec.

4.1 Word2Vec

Word2Vec je kolokvijalni naziv za dva modela *Skip-gram* i *Continious bag-of-words* za učenje embeding vektora riječi koji omogućavaju višestruko procjenjivanje sličnosti, konkretno sintaksnu, ali i semantičku sličnost [16], [8]. *Skip-gram* i *Continious bag-of-words* modeli su varijanta neuralnog jezičnog modela (*Neural Network Language Model*, NNLM) opisanog u [17], no njihov cilj nije jezično modeliranje nego tvorba samih vektora, dok je to kod NNLM-a samo početni korak. Zbog načina na koji se generiraju, vektori riječi koje se pojavljuju u istom kontekstu se grupiraju zajedno u vektorskom prostoru, pa su i dobiveni nisko-

diemnzionalni vektori semantički slični. Tako su Mikolov i suradnici su u [16] pokazali da dobiveni vektori sadrže i semantička svojstva. Na primjer, vektor $vec(\text{"}King\text{"}) - vec(\text{"}Man\text{"}) + vec(\text{"}Woman\text{"})$ je sličan vektoru riječi "Queen", a maksimiziranje točnosti ovog svojstva je jedan od ciljeva učenja modela.

Prvi od dva opisana modela je *Continious bag-of-words (CBOW)*, a razlikuje se od NNLM-a po tomu što u njemu nelinearni prikriveni sloj nije prisutan, te se projekcijski sloj koristi za sve riječi [16]. Neka je $w_t \in V$ ciljana riječ, C veličina "prozora" oko riječi, tj. $w_{t-1}, w_{t-2}, ..., w_{t-C}$ je niz prijašnjih riječi, te $w_{t+1}, w_{t+2}, ..., w_{t+C}$ su riječi koji dolaze nakon w_t, cilj učenja modela je maksimizacija vjerojatnosti [18]:

$$\frac{1}{|V|} \sum_{t=1}^{|V|} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_t | w_{t+j}). \tag{6}$$

Vjerojatnost p se računa softmax funkcijom [18]:

$$p(w_t | w_{t-j}, ..., w_{t+j}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{7}$$

$$y = Uh(w_{t-j}, ..., w_{t+j}; W) + b \tag{8}$$

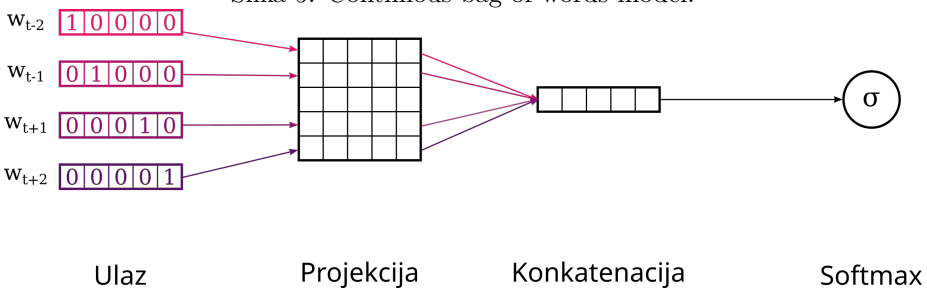
gdje su b i U parametri softmax funkcije, a h je dobiven konkatenacijom ili aritmetičkom sredinom izlaznih vektora iz projekcijskog sloja, tj. matrice W. Matrica W sadrži embedding vektore, a u implementacijama modela vrijednosti tih vektora su najčešće inicijalno postavljene nasumično, te se ažuriraju učenjem kroz algoritam širenja unatrag (*back-propagation*) kako bi se maksimizirala jednadžba 6. Na taj način dolazimo od nasumičnih vektora do onih koji reprezentiraju semantičke odnose u niskodimenzionalnom prostoru. Nadalje, na taj sloj se ne primjenjuje aktivacijska funkcija, a za ulazni *one-hot* vektor x nije zapravo potrebno ni računati produkt xW, već se indeks dimenzije vektora x može koristiti kao indeks retka matrice W. Ovo je jasno iz primjera produkta ulaznog vektora x i embeding matrice W:

$$x = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 0.20 & -1.17 & 2.32 \\ 1.12 & 2.50 & 0.72 \\ -5.27 & 1.13 & -1.00 \\ 0.99 & 0.97 & 1.24 \\ 1.14 & 1.00 & -0.55 \end{bmatrix}$$

Rezultat produkta je vektor $xW = \begin{bmatrix} -5.27 & 1.13 & -1.00 \end{bmatrix}$ budući da dimenzije od x s vrijednošću 0 ne pridonose sumi s općim članovima od W, stoga je moguće zamijeniti ovu operaciju indeksiranjem, te tako smanjiti vremensku kompleksnost programa. Slika 5 shematski prikazuje opisane operacije.

Slika 5: Continious bag-of-words model.

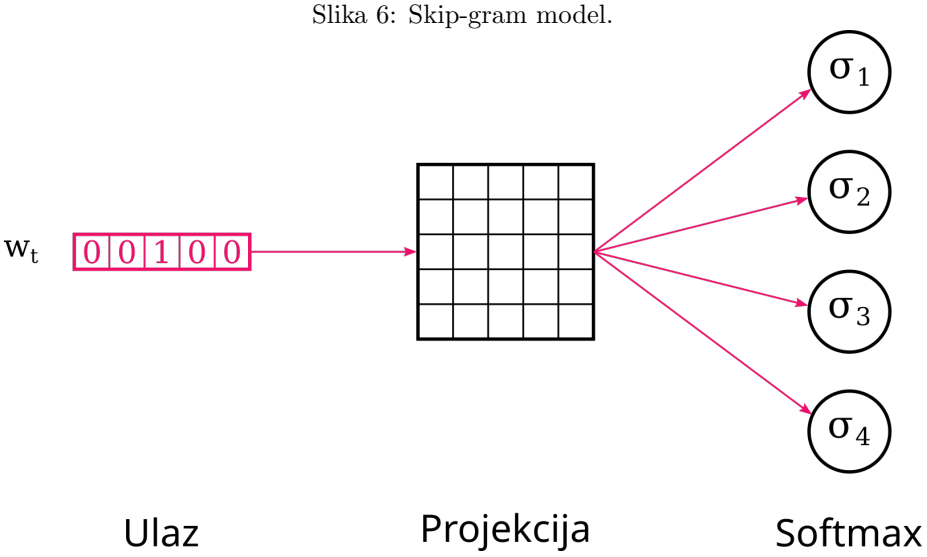


Skip-gram model je vrlo sličan CBOW modelu, te sve dosad rečeno o CBOW modelu vrijedi i za Skip-gram. Međutim, Skip-gram se razlikuje u samoj formulaciji problema, tj. Skip-gram ne nastoji predvidjeti ciljanu riječ iz konteksta, već predvidjeti kontekst za ciljanu riječ [16]. Dakle, za ciljanu riječ w_t i kontekstne riječi $w_{t-C}, w_{t-C+1}, ..., w_{t+C-1}, w_{t+C}$ cilj je maksimizirati vjerojatnost [8]:

$$\frac{1}{|V|} \sum_{t=1}^{|V|} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t). \tag{9}$$

Slika 6 shematski prikazuje Skip-gram model; na ulazu se pojavljuje samo jedna riječ - ciljana riječ, pa nije nužno izvoditi konkatenaciju vektora, odnosno računanje aritmetičke sredine.

Slika 6: Skip-gram model.



Učenje ovih reprezentacija je vremenski skupa operacija koja velikim dijelom ovisi o veličini vokabulara V. Za CBOW model kompleksnost Q iznosi [16]:

$$Q = N \times D + D \times \log_2(V) \tag{10}$$

gdje je N je broj kontekstnih riječi, D dimenzija Word2Vec vektora, odnosno projekcijskog sloja. Logaritamska kompleksnost vokabulara proizlazi iz činjenice da vokabular možemo predstaviti binarnim stablom, te tako smanjiti kompleksnost modela, inače bi zadnji član jednadžbe 10 iznosio $D \times V$ [16]. Za Skip-gram model kompleksnost iznosi [16]:

$$Q = C \times [D + D \times \log_2(V)]. \tag{11}$$

gdje je C veličina kliznog prozora konteksta iz kojeg se tijekom učenja nasumično uzorkuje $R \in [1, C]$ prijašnjih riječi, te R riječi koje slijede ciljanu riječ.

Za smanjenje kompleksnosti softmax funkcija se može aproksimirati s hijerarhijskom softmax funkcijom, te alternativno koristeći tehniku negativnog uzorkovanja (*negative sampling*) [8]. Kada se primjenjuje tehnika negativnog uzorkovanja, učenje vektorskih reprezentacija se postiže binarnom klasifikacijom, odnosno umjesto predviđanja kontekstnih riječi u izvornome Skip-gram modelu, reprezentacije se uče klasifikacijom parova riječi koji se pojavljuju, odnosno ne pojavljuju zajedno u kontekstu [8]. Na ovaj način, problem klasifikacije u V klasa se zamjenjuje binarnom klasifikacijom, pa se složena softmax funkcija zamjenjuje jednostavnijom sigmoid funkcijom, npr. logističkom funkcijom (*logistic function*) $\sigma = \frac{1}{1+e^{-x}}$. Cilj učenja modela tada postaje [8]:

$$\log \sigma(v'^{\top}_{w_O} v_{w_I}) + \sum_{i=1}^{k} E_{w_i} \sim P_n(w)[\log \sigma(-v'^{\top}_{w_i} v_{w_I})] \tag{12}$$

gdje je w_o je ciljana riječ, w_I kontekstna riječ, a w_i negativni uzorak (*negative sample*), tj. riječi koje nisu u kontekstu ciljane riječi. Model daje najbolje rezultate kada je k negativnih uzoraka uzorkovano iz unigram distribucije $P_n = \frac{U_n(w)^{\alpha}}{Z}$, gdje je α hiperparameter za koji se eksperimentalno pokazalo da daje najbolje rezultate za $\alpha = \frac{3}{4}$ [8].

4.2 Doc2Vec

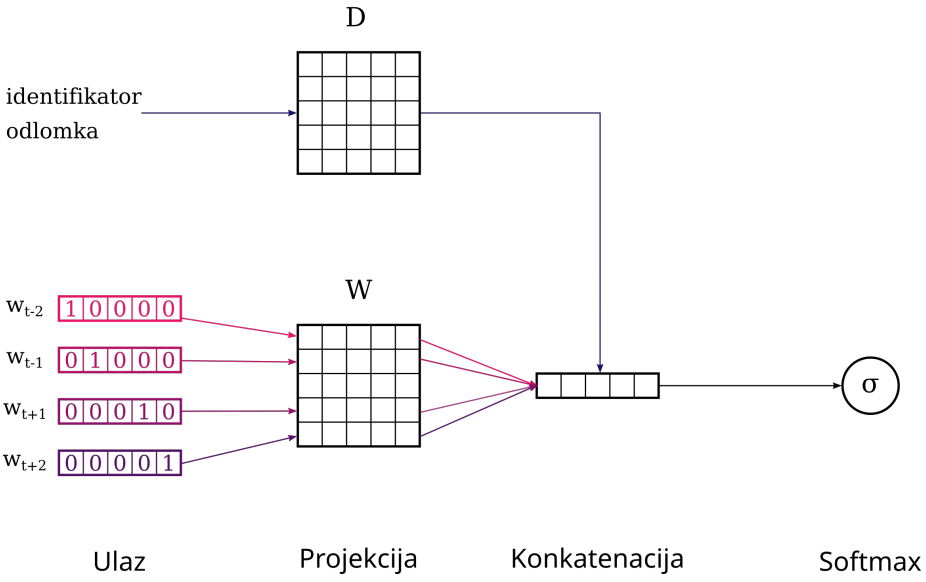
Word2Vec je koristan alat za vektorsku reprezentaciju individualnih riječi, ali nije primjeren za dulje tekstove. Ukoliko želimo Word2Vec-om predstaviti dulji tekst, poput rečenice ili cijelog dokumenta, najčešće se računa aritmetička sredina vektora svih riječi, npr. aritmetička sredina svih riječi u rečenici. Na ovaj način dobije se jedan vektor koji djelomično obuhvaća značenje svih riječi u tom tekstu, tj. centralni vektor teksta koji predstavlja svojevrstan "semantički prosjek" teksta.

Bolji način za reprezentaciju duljih tekstova dobiva se Doc2Vec modelom [18]. Iako se Doc2Vec algoritam za tvorbu vektora zove vektor odlomka (*paragraph vector*, PV), on se može primijeniti na proizvoljno dugi dio tekst, dakle

rečenicu, odlomak, cijeli dokument i slično [18]. Poput Word2Vec-a, Doc2Vec također sadrži dva modela za tvorbu vektora koji se baziraju na vektoru odlomka - memorijski vektor odlomka (*paragraph vector - distributed memory, PV-DM*), te distribuirani vektor odlomka (*paragraph vector - distributed bag-of-words, PV-DBOW*) [18].

Prvi opisani model, tj. onaj korišten u ovome radu je PV-DM. PV-DM je vrlo sličan CBOW modelu čiji je zadatak predvidjeti ciljanu riječ iz kontekstnih riječi. Uz vektore kontekstnih riječi, vektor odlomka također igra ulogu u predviđanju. Kroz učenje modela vektor odlomka dolazi od nasumičnih vrijednosti do onih koje izražavaju određenu semantiku poput CBOW vektora [18]. Slika 7 prikazuje shemu PV-DM modela. Kao i kod CBOW modela, vektori kontekstnih riječi se preslikavaju na redove matrice W koji se zatim međusobno konkateniraju. Razlika između CBOW i PV-DM modela je što PV-DM dodatno sadrži i matricu D u kojoj su zapisani embeding vektori odlomka koji se konkateniraju s vektorima riječi [18]. Cilj učenja je i dalje maksimizirati jednadžbu 6, međutim h iz jednadžbe 8 se dobiva konkatenacijom vektora iz matrice W s vektorom iz matrice D [18].

Slika 7: Paragraph vector - distributed memory model.



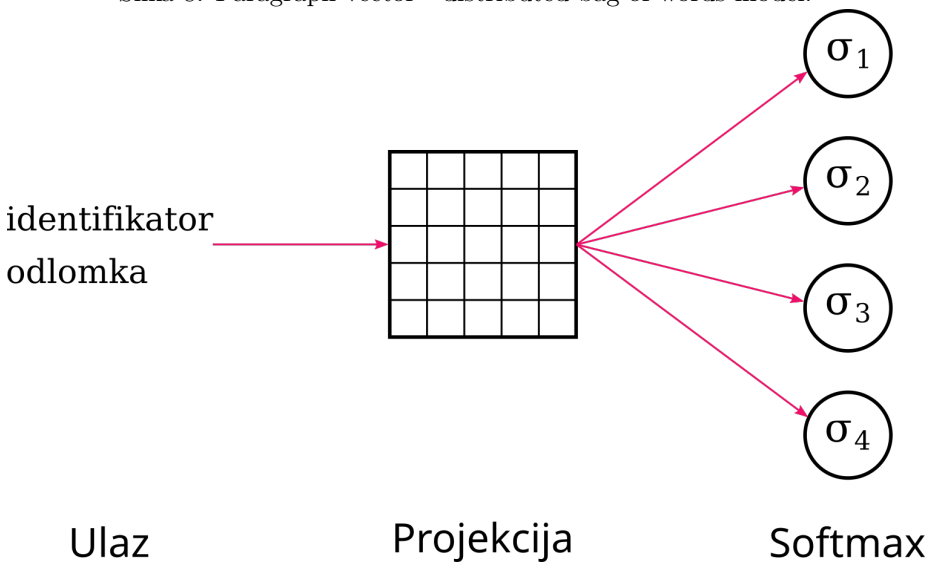
Vektori riječi se mijenjaju dok prozor klizi kroz odlomak, a sam vektor odlomka ostaje isti [18]. Tek kada u učenje ulazi sljedeći odlomak, mijenja se i vektor iz D. Dakle, vektor odlomka "pamti" ono što nedostaje u trenutnom kontekstu budući da je on prisutan u svim kontekstima, pa je ovo i razlog njegovog imenovanja. Riječi, odnosno vektori iz W su isti za svaki odlomak, npr. riječ "rad" se preslikava na isti red matrice W neovisno o odlomku čija se reprezentacija uči u danom trenutku.

Drugi model, DBOW, više sliči na Skip-gram model iz Word2Vec-a, tj. na ulazu nema kontekstne riječi već je cilj modela predvidjeti kontekstne riječi na temelju ulaznog odlomka [18]. Dakle jednadžba 9 tada postaje:

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|d\in D) \tag{13}$$

gdje je $d \in D$ vektor odlomka, tj. red matrice D. Slika 8 prikazuje shemu DBOW modela; u svakom koraku algoritma uzima se jedna riječ iz klizajućeg prozora te se zatim ona klasificira na temelju ulaznog vektora odlomka [18].

Slika 8: Paragraph vector - distributed bag-of-words model.



5 Podatkovni skup

Za učenje i evaluaciju metoda za sumarizaciju korišten je CNN/DailyMail skup podataka originalno namijenjen za rješavanje problema traženja odgovora na postavljena pitanja (*question answering*) [19], ali se koristi i za evaluaciju modela za ekstraktivnu i apstraktivnu sumarizaciju [20], [21], [22]. Podatkovni skup se sastoji od novinskih članaka prikupljenih sa CNN i DailyMail portala, a svaki članak je popraćen s istaknutim rečenicama koje se ne pojavljuju u izvornom tekstu, ali ističu najvažnije stvari iz svakog članka, pa time tvore i sažetak svakog članka. Tablica 1 prikazuje statistiku[1] CNN i DailyMail korpusa.

Tablica 1: Statistika korpusa.

Statistika	CNN	DailyMail
Broj članaka	92 579	219 506
Broj pojavnica (tokena)	75 271 747	185 267 371
Ukupan broj rečenica	3 162 580	5 974 954
Prosječni broj rečenica	34.16	27.22
Prosječni broj rečenica u sažetku	3.56	3.83

Za učenje korištenih modela koristio se CNN dio korpusa koji je podijeljen u omjeru 80:20 na dio za učenje i dio za testiranje. Dio za učenje je dodatno podijeljen u omjeru 90:10 na dio za učenje i validaciju parametara neuronske mreže. Cijeli CNN korpus se također koristio za učenje Word2Vec, odnosno Doc2Vec reprezentacija.

6 Evaluacija

Evaluacija postupaka sumarizacije zasniva se na usporedbi automatski generiranog sadržaja s referentnim tekstom kojeg je napisao (odredio) čovjek. Dakle evaluacija se provodi s ciljem da se računalno generiran sadržaj čim više približi referentnom sadržaju. Najčešće korištena mjera evaluacije je ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) koja se koristi i u ovome radu. ROUGE je mjera koja mjeri preklapanja n-grama, tj. uzastopnih nizova od n riječi između generiranog sažetka, te referentnih sažetaka koji se uzimaju kao zlatni standard, te na taj način kvantificira kvalitetu generiranog sažetka [24]. ROUGE-n se definira kao odziv (*recall*) n-grama između generiranog sažetaka S_c i N referentnih sažetka S_i [24] [25]:

$$ROUGE - n(S_c, S_i) = \frac{\sum_{i=1}^{N} |S_c \cap S_i|}{\sum_{i=1}^{N} |S_i|} \tag{14}$$

Svaki članak iz korištenog podatkovnog skupa je popraćen samo jednim sažetkom, pa se jednadžba može pojednostaviti:

[1] U tablici 1 navedene su vrijednosti dobivene postupkom nenadziranog učenja razdjeljivanja (tokenizacije) rečenica iz [23], te se donekle razlikuju od statistika iz [19].

$$ROUGE - n(S_c, S_r) = \frac{|S_c \cap S_r|}{|S_r|} \tag{15}$$

gdje je S_c skup n-grama iz generiranog sažetka, a S_r skup n-grama iz referentnog sažetka.

U ovom radu su korištene ROUGE mjere za $n = 1$, i $n = 2$, te ROUGE-LCS mjera. ROUGE-LCS mjera kvantificira preklapanje n-grama na razini nizova riječi, odnosno preklapanje n-grama između najduljih zajedničkih podnizova [24]. Budući da ROUGE-LCS radi s nizovima, ta mjera uzima u obzir poredak riječi, te tako uzima u obzir strukturu rečenice na prirodan način [24]. ROUGE-LCS se temelji na F_β mjeri, odnosno njenoj pojednostavljenoj F1 mjeri te se definira kao [24]:

$$F_{LCS}(S_c, S_r) = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{16}$$

$$R_{LCS}(S_c, S_r) = \frac{LCS(S_c, S_r)}{|S_r|} \tag{17}$$

$$P_{LCS}(S_c, S_r) = \frac{LCS(S_c, S_r)}{|S_c|} \tag{18}$$

$LCS(S_c, S_r)$ dužina najdužeg zajedničkog podniza sažetaka S_c i S_r, a R_{LCS} i P_{LCS} odgovaraju odzivu i preciznosti. Za računanje ROUGE vrijednosti u radu je korišten ROUGE 2.0 programski paket [26].

Točnost klasifikacije A (*accuracy*) je definirana kao [25]:

$$A = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \tag{19}$$

gdje su TP, TN, FP, FN redom skupovi stvarno pozitivnih, stvarno negativnih, pogrešno pozitivnih te pogrešno negativnih klasificiranih primjera, odnosno rečenica.

7 Implementacija

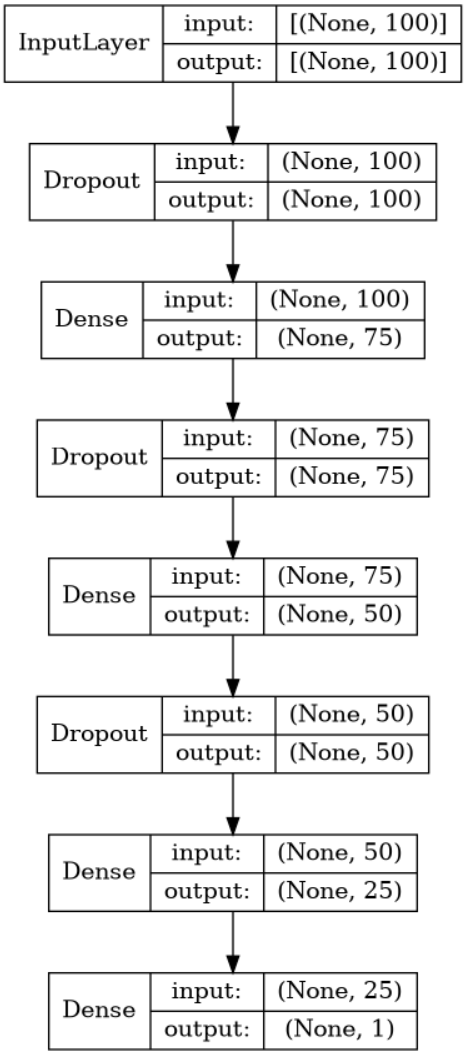
7.1 Učenje reprezentacija

Za učenje Word2Vec i Doc2Vec reprezentacija korišten je Gensim alat [27]. Reprezentacije su učene na cijelom CNN dijelu korpusa koji je pretprocesiran neposredno prije učenja na način da su izbačene zaustavne riječi (*stopwords*), znamenke te interpunkcijski i ostali nepotrebni znakovi. Za razdjeljivanje rečenica korištena je NLTK [28] implementacija algoritma opisanog u [23] koji je prilagođen na DailyMail dijelu korpusa. Za Word2Vec reprezentaciju učeni su 100-dimenzionalni CBOW vektori na primjerima rečenica iz CNN članaka kroz 400 epoha koristeći tehniku negativnog uzorkovanja (*negative sampling*) za $k = 5$ negativnih uzoraka i parametrom $\alpha = \frac{3}{4}$. U slučaju Doc2Vec modela, naučene su PV-DM reprezentacije individualnih rečenica iz članaka s istim parametrima kao i kod Word2Vec modela, a za dobivanje vektora h iz jednadžbe 8 umjesto konkatenacije, izračunata je suma vektora. U implementaciji oba modela veličina kliznog prozora iznosi 5.

7.2 Višeslojni perceptron

Višeslojni perceptron je implementiran koristeći biblioteku Keras [29], a zamišljen je kao binarni klasifikator koji klasificira rečenice u binarnu klasu ovisno pripada li ona sažetku ili ne. Model se sastoji od 4 sloja, tri prikrivena i jedan izlazni. Na ulaz se dovode rečenice reprezentirane 100-dimenzionalnim CBOW vektorima koji su naučeni unaprijed. Prvi prikriveni sloj se sastoji od 100 neurona, zatim drugi od 75 neurona, te konačno zadnji prikriveni sloj se sastoji od 50 neurona. Na sva tri skrivena sloja primjenjena je ReLU aktivacijska funkcija $f(x) = max(0, x)$. Prije svakog potpuno povezanog (*dense*) sloja primjenjen je dropout koji isključuje određene neurone ovisno o zadanoj frekvenciji, konkretno u ovom slučaju ona iznosi 0.25. Na ovaj način dropout pomaže smanjiti mogućnost pretjeranog prilagođavanja vrijednostima podataka za učenje (*overfitting*) [7]. Izlaz iz zadnjeg sloja je realan broj izračunat pomoću logističke funkcije, te predstavlja vjerojatnost da rečenica pripada sažetku. Slika 9 prikazuje implementacijske slojeve Kerasa za višeslojni perceptron.

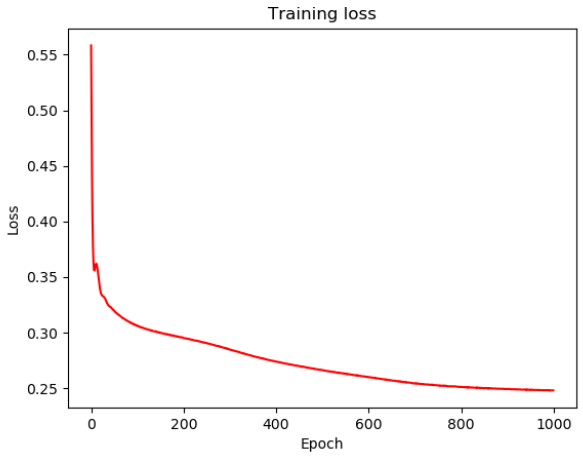
Slika 9: Implementacijski slojevi višeslojnog perceptrona.



Za funkciju gubitka uzeta je binarna unakrsna entropija (*binary crossentropy*), a kao optimizacijski algoritam korišten je Adam [30]. Model je učen kroz 1000 epoha. Tijekom klasifikacije rečenice se rangiraju prema predviđenoj vjerojatnosti od najveće do najmanje, a zatim se uzme prvih N rečenica ovisno o tome želimo li uključiti 1, 3, 5 ili 10 rečenica u sažetak. Za potrebe ovog rada uzete su prve 3 rečenice što odgovara zadatku TOP 3 izlučivanja. Ovdje je TOP 3 zadatak odabran zbog prosječnog broja rečenica u sažetku koji iznosi 3.56 (naveden u tablici 1).

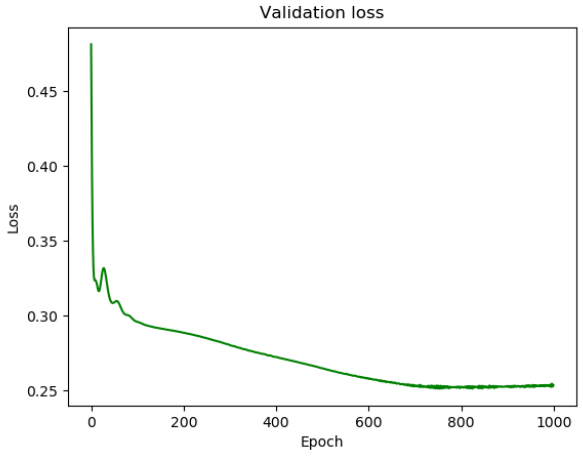
Slika 10 prikazuje graf funkcije gubitka tijekom 1000 epoha učenja MLP modela. Funkcija gubitka se smanjila sa 0.55 na 0.25. Graf funkcije gubitka ukazuje da je model kontinuirano učio i popravljao parametre sve do zadnje epohe.

Slika 10: Dijagram funkcije gubitka tijekom učenja MLP modela.



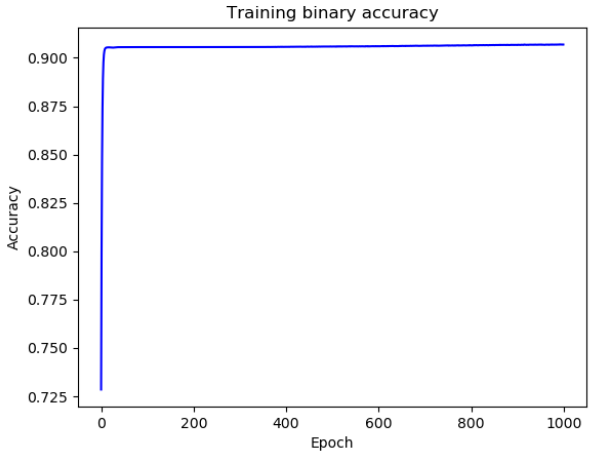
Slika 11 prikazuje graf funkcije gubitka za validacijski skup podataka tijekom 1000 epoha učenja MLP modela. Grafovi funkcije gubitka na skupu za učenje i skupu za validaciju ukazuju na konzistentno učenje parametara modela, te se iz njega vidi da model nije pretjerano prilagođen podacima za učenje.

Slika 11: Dijagram funkcije gubitka tijekom validacije MLP modela.



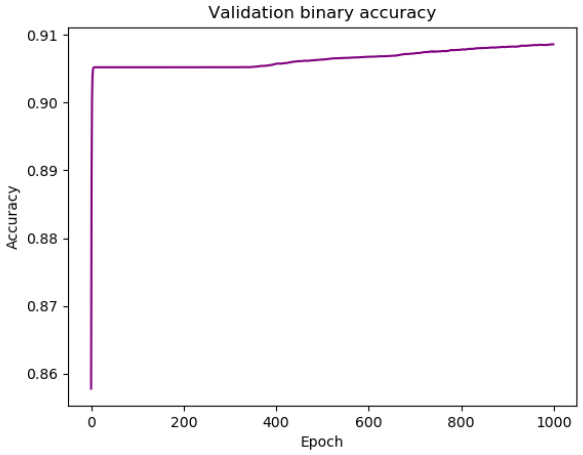
Slika 12 prikazuje binarnu točnost klasifikacije tijekom učenja MLP modela. Na slici se vidi da MLP model već od 1 epohe točno klasificira gotovo sve rečenice označene kao sažetak.

Slika 12: Dijagram binarne točnosti tijekom učenja MLP modela.



Slika 13 prikazuje binarnu točnost klasifikacije tijekom validacije MLP modela. Također, iz grafa je razvidno da je MLP model početno imao blage nepreciznosti u klasifikaciji rečenica iz validacijskog skupa, ali su one ispravljene nakon približno 400 epoha. Slike 12 i 13 služe kao kontrolna vizualizacija postupka učenja.

Slika 13: Dijagram binarne točnosti tijekom validacije MLP modela.

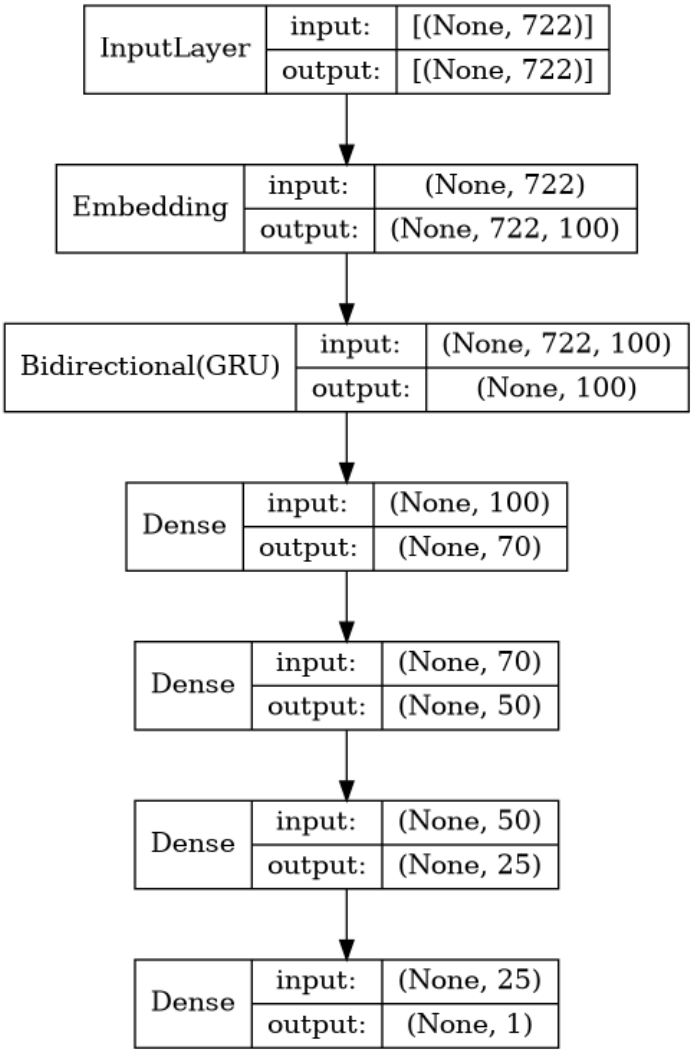


7.3 Upravljačka rekurentna jedinica

Poput višeslojnog perceptrona, GRU model također klasificira ulazne rečenice u binarnu klasu koja označava pripadnost sažetku, no umjesto klasifikacije Doc2Vec vektora ova arhitektura klasificira prikrevena stanja GRU-a. Ulaz u model su nizovi riječi dugi do 722 riječi, tj. vektori čije se dimenzije presklikavaju na redove embedding matrice, tj. embeding vektore dimenzija 100. Embeding matrica je incijalizirana sa unaprijed naučenim CBOW vektorima koji postaju ulaz u sljedeći sloj. Dropout se primjenjuje na ulazne vektore u GRU s frekvencijom 0.25, a nizovi se obrađuju u oba smjera, te se vektori izlaznih stanja konkateniraju. Dakle, GRU uči prikrivena stanja s 50 dimenzija kada obrađuje niz u jednom smjeru, pa nakon konkatenacije s vektorom naučenog iz niza suprotnog smjera, vektor prikrivenog stanja je dimenzije 100. Izlazno stanje se zatim klasificira na potpuno identičan način kao kod višeslojnog perceptrona opisanog iznad, dakle prikriveno stanje postaje ulaz u niz od tri potpuno povezana sloja s ReLU aktivacijom, te je izlaz vjerojatnost klase dobivena logističkom funkcijom koja se primjenjuje na izlazni sloj. Još jedan način na koji se ovo može gledati je da izlaz iz GRU-a postaje ulaz u model višeslojnog perceptrona koji nema dropout prije potpuno povezanih slojeva budući da je on već primjenjen na ulaz u GRU.

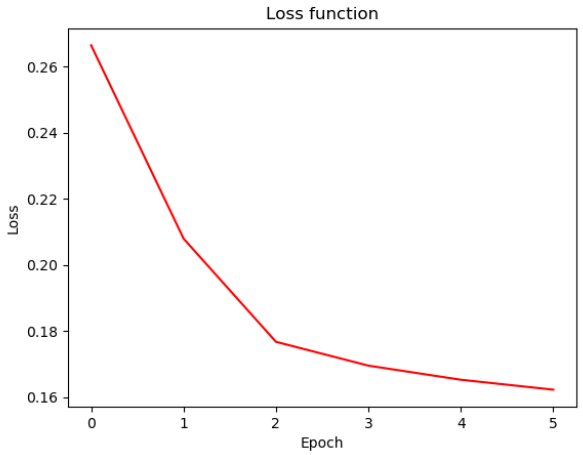
GRU model je učen u paketima (*batches*) od 180 dokumenata kroz 15 epoha, ali uz prijevremeno zaustavljanje (*early stopping*) ako ne uspije smanjiti funkciju gubitka u jednoj epohi učenja ili povećati binarnu točnost (*binary accuracy*) u 3 epohe. U provedenom eksperimentu, model se zaustavio nakon 5 epoha. Slika 14 prikazuje implementacijske slojeve Kerasa za GRU.

Slika 14: Implementacija GRU arhitekture.



Slika 15 prikazuje graf funkcije gubitka tijekom 5 epoha učenja GRU modela. Funkcija gubitka se smanjila sa 0.26 na 0.16, te je znatno brže konvergirala nego funkcija gubitka MLP modela.

Slika 15: Dijagram funkcije gubitka tijekom učenja GRU modela.

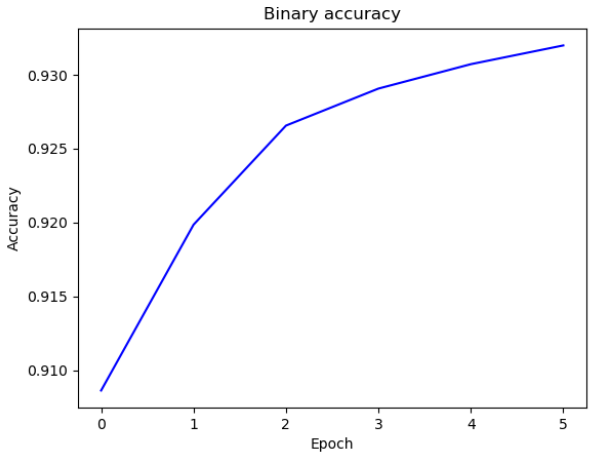


Slika 16 prikazuje graf funkcije gubitka za validacijski skup podataka tijekom 5 epoha učenja GRU modela. Grafovi funkcije gubitka na skupu za učenje i skupu za validaciju ukazuju na konzistentno učenje parametara modela kao što je to slučaj kao i kod MLP modela.

Slika 16: Dijagram funkcije gubitka tijekom validacije GRU modela.

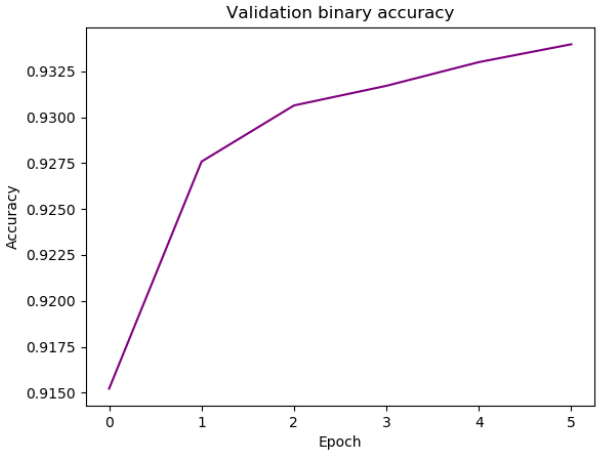
Slika 17 prikazuje binarnu točnost klasifikacije tijekom učenja GRU modela. Na slici se vidi da GRU model također od 1. epohe točno klasificira gotovo sve rečenice označene kao sažetak, međutim u odnosu na MLP inicijalna točnost klasifikacije je veća, pa je povećanje točnosti manje.

Slika 17: Dijagram binarne točnosti tijekom učenja GRU modela.



Slika 18 prikazuje binarnu točnost klasifikacije tijekom validacije GRU modela. Rast točnosti klasifikacije tijekom validacije donekle prati rast točnosti tijekom učenja, te je razvidno da ni GRU model nema problema s pretjeranom prilagodbom podacima za učenje.

Slika 18: Dijagram binarne točnosti tijekom validacije GRU modela.



8 Rezultati

U Tablicama 2, 3 i 4 prikazani su rezultati za odziv, preciznost i $F1$ mjeru. Rezultati prikazuju performanse MLP i GRU modela za ROUGE-1, ROUGE-2 i ROUGE-LCS mjeru na zadatku sažimanja TOP 3 rečenice.

Tablica 2: Rezultati sumarizacije - odziv.

Model	ROUGE-1	ROUGE-2	ROUGE-LCS
MLP	**0.2843**	**0.0803**	**0.2581**
GRU	0.1979	0.0692	0.1874

Tablica 3: Rezultati sumarizacije - preciznost.

Model	ROUGE-1	ROUGE-2	ROUGE-LCS
MLP	0.2371	0.0636	0.2211
GRU	**0.3140**	**0.1030**	**0.2773**

Tablica 4: Rezultati sumarizacije - F1.

Model	ROUGE-1	ROUGE-2	ROUGE-LCS
MLP	**0.2527**	0.0691	**0.2339**
GRU	0.2307	**0.0783**	0.2135

Prema odzivu iz tablice 2 perceptron MLP postiže bolje vrijednosti od GRU modela, dok GRU ima superiorniju preciznost (tablica 3) dok su prema F1 mjeri podjednaki (tablica 4).

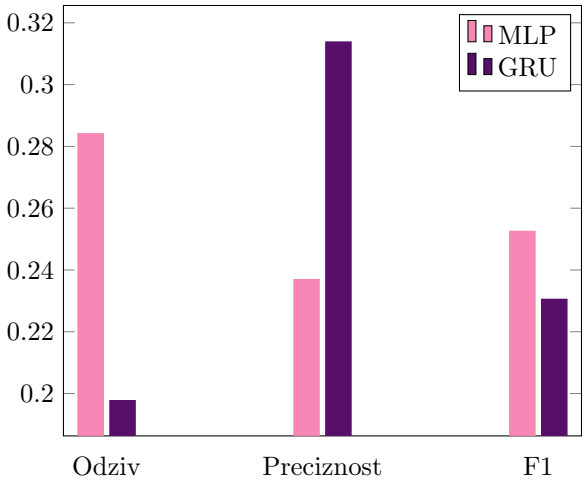
Na slikama 19, 20 i 21 vizualizirani su rezultati za ROUGE-1, ROUGE-2 i ROUGE-LCS za usporedbu MLP i GRU modela po odzivu, preciznosti i F1 mjeri. Iz slika je razvidno da MLP model ima visoki odziv bez obzira na način evaluacije, ali zbog smanjene preciznosti (odnosno prevelikog broja rečenica koje klasificira u sažetak) ne postiže uvijek visoku vrijednost F1 mjere. Preveliki broj rečenica koje MLP smatra sažetkom uključuju i točne rečenice, ali visoki odziv nije poželjna osobina modela u slučaju kad želimo dobivati koncizni informativni sadržaj, kao što je to u slučaju sumarizacije.

Stoga je GRU, koji postiže bolju preciznost ekstrakcije primjerenija metoda za korištenje kada želimo koncizan sadržaj, posebno sa stanovišta postizanja dobrih rezultata prema striktnoj ROUGE-2 evaluaciji, a MLP ukoliko želimo više rečenica u sažetku.

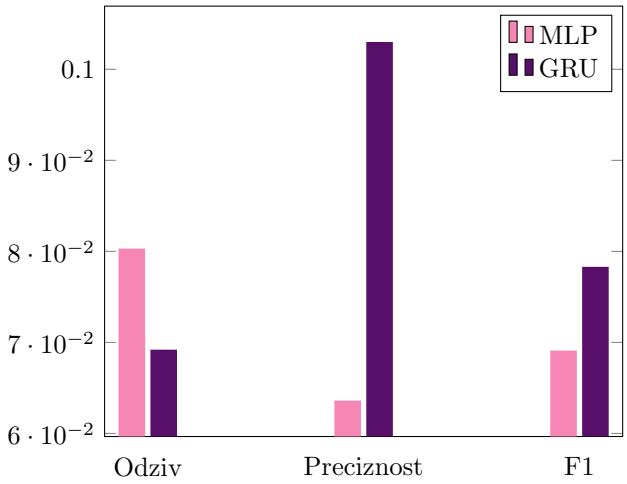
Mjere ROUGE-1 i ROUGE-LCS su manje striktne za evaluaciju, dok ROUGE-2 striktnije procjenjuje sažetak jer uvažava isključivo preklapanja bigrama. ROUGE-1 i ROUGE-LCS na ovom skupu podataka postižu slične vrijednosti. ROUGE-LCS je možda najprimjernija mjera za evaluaciju jer procjenjuje presjek najduljih zajedničkih podnizova, ali pri tome ne uvažava udaljenost između pojedinih riječi u nizu - dakle ubacivanje ne smatra pogreškom.

Zaključno, iako MLP i GRU postižu podjednake rezultate za jednostavan zadatak ekstraktivne sumarizacije, bolji ROUGE-2 rezultat sugerira da je GRU potencialno primjereniji za primjenu u kompleksnijim zadacima poput abstraktivne sumarizacije.

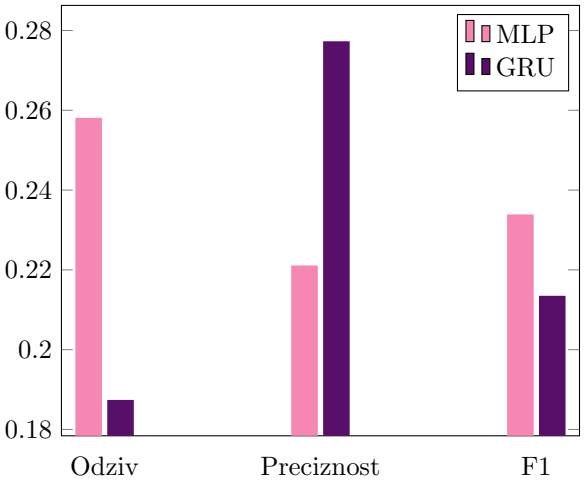
Slika 19: Rezultati sumarizacije - ROUGE-1



Slika 20: Rezultati sumarizacije - ROUGE-2



Slika 21: Rezultati sumarizacije - ROUGE-LCS.



9 Zaključak

U diplomskom radu su predstavljeni postupci ekstraktivnog sažimanja tekstualnih dokumenata metodama višeslojnog perceptrona i GRU rekurentne duboke mreže. Oba modela su učeni kao binarni klasifikatori koji određuju pripada li rečenica sažetku ili ne. U višeslojnom perceptronu se učenje temelji na Doc2Vec reprezentacijama rečenica, dok se u GRU reprezentacija uči iz originalnih tekstova. Vrijednosti težina u prvom prikrivenom sloju - embeding sloju - u GRU modelu su inicijalizirani vrijednostima Word2vec vektora za vokabular korištenog skupa podataka. Modeli su naučeni i testirani na CNN/DailyMail skupu podatka. Performanse naučenih modela procijenjene su standardnim postupcima evaluacije sažimanja teksta ROUGE-1, ROUGE-2 i ROUGE-LCS mjerama koje su računate za odziv, preciznost i F1 mjeru na zadatku ekstrakcije TOP 3 riječi.

Generalno, višeslojni perceptron postiže bolje rezultate ukoliko se uvažava odziv, dok GRU postiže preciznije rezultate bez obzira na ROUGE mjeru evaluacije. U sustavima gdje je preciznost važnija osobina od odziva, GRU je primjereniji metoda, dok je višeslojni perceptron primjereniji za probleme u kojima je odziv jednako važan ili važniji.

Iako daleko od mogućnosti tvorenja optimalnih sažetka, rezultati ukazuju na primjerenost ovih metoda za problem ekstraktivne sumarizacije teksta. Glavni problem dubokog učenja je dovoljna količina podataka, koja se pokazala ključnom i na ovom jednostavnom problemu binarne klasifikacije.

U ovom radu primijenjene su osnovne metode za ekstraktivno sažimanje teksta koje se mogu proširiti cijelim nizom state-of-the-art metoda za koje je poznato da postižu bolje rezultat, ali isto tako su računski zahtjevne poput SummaRunner [20] ili Pointer Generator dubokih mreža [21]. Naposljetku jedan od izazova u budućem radu je i učenje abstraktivnih modela za sumarizaciju.

Literatura

[1] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, "An overview of graph-based keyword extraction methods and approaches," *Journal of information and organizational sciences*, vol. 39, no. 1, pp. 1–20, 2015.

[2] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, "Selectivity-based keyword extraction method," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 12, no. 3, pp. 1–26, 2016.

[3] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[4] R. Mishra, J. Bian, M. Fiszman, C. R. Weir, S. Jonnalagadda, J. Mostafa, and G. Del Fiol, "Text summarization in the biomedical domain: a systematic review of recent research," *Journal of biomedical informatics*, vol. 52, pp. 457–467, 2014.

[5] J. L. Neto, A. A. Freitas, and C. A. Kaestner, "Automatic text summarization using a machine learning approach," in *Brazilian symposium on artificial intelligence*, pp. 205–215, Springer, 2002.

[6] D. Aljević, L. Todorovski, and S. Martinčić-Ipšić, "Extractive text summarization based on selectivity ranking," in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–6, IEEE, 2021.

[7] Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Mar. 2017.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," p. 9, Dec. 2013.

[9] K. Babić, S. Martinčić-Ipšić, and A. Meštrović, "Survey of neural text representation models," *Information*, vol. 11, no. 11, p. 511, 2020.

[10] G. Carenini, G. Murray, and R. Ng, "Methods for mining and summarizing text conversations," *Synthesis Lectures on Data Management*, vol. 3, no. 3, pp. 1–130, 2011.

[11] M. Minsky and Seymour Papert, *Perceptrons: An Introduction to Computational Geomtry*. 1969.

[12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[13] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," *arXiv:1211.5063 [cs]*, Feb. 2013. arXiv: 1211.5063.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[15] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv:1406.1078 [cs, stat]*, Sept. 2014. arXiv: 1406.1078.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781 [cs]*, Sept. 2013. arXiv: 1301.3781.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model," p. 19, 2003.

[18] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," p. 9, 2014.

[19] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching Machines to Read and Comprehend," *arXiv:1506.03340 [cs]*, Nov. 2015. arXiv: 1506.03340.

[20] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents," *arXiv:1611.04230 [cs]*, Nov. 2016. arXiv: 1611.04230.

[21] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1073–1083, Association for Computational Linguistics, 2017.

[22] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," *arXiv:1912.08777 [cs]*, July 2020. arXiv: 1912.08777.

[23] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," *Computational Linguistics*, vol. 32, pp. 485–525, Dec. 2006.

[24] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," p. 8, July 2004.

[25] L. Ermakova, J. V. Cossu, and J. Mothe, "A survey on evaluation of summarization methods," *Information Processing & Management*, vol. 56, no. 5, pp. 1794–1814, 2019.

[26] K. Ganesan, "ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks," *arXiv:1803.01937 [cs]*, Mar. 2018. arXiv: 1803.01937.

[27] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

[28] S. Bird, Ewan Klein, and Edward Loper, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[29] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.