QA sistem zasnovan na sličnosti pitanja i rečenice u kojoj se nalazi odgovor

Cvijetin Mlađenović
Fakultet tehničkih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6
21000 Novi Sad
mladjenovic.cvijetin@gmail.com

Marko Radović
Fakultet tehničkih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6
21000 Novi Sad
markoradovic996@gmail.com

Borislav Gajić
Fakultet tehničkih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6
21000 Novi Sad
borawr22@gmail.com

***Apstrakt*— Proširenost interneta poslednjih decenija je omogućila generisanje velike količine podataka koju je veoma izazovno pretražiti na dovoljno brz i semantički način. U ovom radu su istražene različite tehnike određivanja semantičke sličnosti između pitanja i rečenice u kojoj se potencijalno nalazi odgovor na pitanje. Cilj je odrediti da li izjavna rečenica sadrži odgovor na postavljeno pitanje ili ne. Kao skup podataka za trening i evaluaciju je korišćen WikiQA. Korišćena su tri osnovna modela: *Convolutional Neural Network* (CNN), *Recurent Neural Network* (RNN) i transformer. U kombinaciji sa CNN i RNN su isprobani kao reprezentativni (engl. *embedding)* slojevi pretrenirani *GloVe* vektori. Pretrenirani transformer model je korišćen u okviru *fine-tuning* pristupa. Kao metrika za evaluaciju performansi modela je korišćena makro F1 mera zbog velike nebalansiranosti ciljne labele. Konačno, na testnom skupu podataka RNN postiže 0.54, CNN 0.58 a transformer 0.74 makro F1 meru.**

Keywords*— *QA, CNN, RNN, transformer, neural network

I. Uvod

Eksponencijalni rast javno dostupnih podataka predstavlja izazov za kvalitetnu i efikasnu pretragu te količine podataka. Takođe, mnoge firme poseduju rastuću količinu dokumenata koju je neophodno pretražiti. Do sada aktuelni statistički modeli pretrage su se pokazali dosta efikasno, ali imaju ograničenje u mogućnosti razumevanja semantike postavljenog upita. Neuronske mreže su se pokazale kao dobro rešenje za semantičke probleme nad kraćim tekstovima. Ovaj rad će pokušati da reši problem pronalaska odgovora na pitanje na semantički način, ali tako da se rešenje može iskoristiti za veće količine dokumenata.

U ovom radu će biti predstavljeno par rešenja problema predikcije sličnosti odgovora na postavljeno pitanje QA (*question answering*). Za obučavanje modela koriste se parovi pitanja i odgovora te se moraju primeniti razne tehnike obrade teksta pre samog obučavanja. Za sam model rešenje može obuhvatati par pristupa, ali će ovaj rad biti fokusiran na neuronske mreže specijalizovane za obučavanje nad tekstualnim podacima NLP (*Natural Language Processing)*.

Konkretne metodologije koje su korišćene u radu su: rekurentna i konvolutivna neuronska mreža sa pretreniranom *GloVe B6* reprezentacijom. Pored navedenih isprobana su i dva pretrenirana transformer modela, gde je jedan dodatno treniran za semantičke sličnosti a drugi semantičke pretrage.

Najviše uspeha ostvareno je upotrebom transofrmera, dok su nešto lošiji rezultati dobijeni korišćenjem RNN i CNN mreža.

Detaljan opis skupa podataka, metoda pomoću kojih je problem rešavan i izazovi prilikom obučavanja modela biće predstavljeni u nastavku rada. U narednom poglavlju su ukratko izložena srodna istraživanja, odnosno radovi sa sličnom temom i metodikom. U trećem poglavlju biće predstavljen skup podataka koji je korišćen kao i način pripreme podataka za obučavanje i testiranje modela. Zatim se predstavlja metodologija koja se koristi za rešavanje QA problema slicnosti pitanja i odgovora. Nakon toga sledi prikaz rezultata sa različitim pristupima. Na kraju je predstavljen zaključak ovog rada.

II. Srodna istraživanja

A. Rad "*WIKIQA: A Challenge Dataset for Open-Domain Question Answering*" [1] se bavi problemom kreiranja skupa podataka (pitanja i odgovora) zarad unapređenja u istraživanjima sistema pitanja i odgovora. Skup pitanja je formiran uz pomoć sajta za pretragu Bing dok su odgovori nastali kao referenca na sajt wikipedia koji bi odgovarao postavljenom pitanju. Odgovori su izabrani po relevantnosti unutar sekcije rezimea. Radi filtriranja relevantnosti datog odgovora na postavljeno pitanje korišćena je grupa ljudi koja je ručno filtrirala odgovore. Na kraju je u skup podataka ušlo 3047 pitanja i 29258 rečenica, gde je 1473 rečenica označeno kao odgovor na dato pitanje.

Do sada je važila pretpostavka da u sistemima koje rešavaju problem nalaženja odgovora na postavljeno pitanje, svako pitanje mora imati barem jedan tačan odgovor. Ovaj rad uvodi novi izazov gde je cilj pronaći da li postoji tačan odgovor u listi kandidata za dato pitanje i vratiti tačan odgovor ako postoji.

Tabela 1 nudi prikaz rezultata modela korišćenjem metoda PV (*Paragraph vector*) i CNN u poređenju WikiQA i QASent skupa podataka koji je predstavljao prethodni *state of the art* skup za QA problem. MRR (*Mean Reciprocal Rank*) i MAP (*Mean Average Precision*) predstavljaju metrike za evaluaciju koje su odabrane u ovom radu.

Model	QASent		WikiQA	
	MAP	MRR	MAP	MRR
Word Cnt	0.5919	0.6662	0.4891	0.4924
Wgt Word Cnt	0.6095	0.6746	0.5099	0.5132
LCLR	**0.6954**	0.7617	0.5993	0.6086
PV	0.5213	0.6023	0.5110	0.5160
CNN	0.5590	0.6230	0.6190	0.6281
PV-Cnt	0.6762	0.7514	0.5976	0.6058
CNN-Cnt	0.6951	**0.7633**	**0.6520**	**0.6652**

Tabela 1. Prikaz dobijenih rezultata u radu "*WIKIQA: A Challenge Dataset for Open-Domain Question Answering*".

B. U radu "*ReQA: An Evaluation for End-to-End Answer Retrieval Models*" [2] su autori uporedili više modela za problem odgovaranja na pitanja iz opšteg domena (engl. *open domain question answering*). Prevashodno, rad se koncentriše na problem pronalaska odgovora na pitanje među velikom kolekcijom dokumenata. Do tada je većina modela pravljena za problem pronalaska tačnog odgovora unutar manjeg korpusa dokumenata. Ovaj problem je sličan *information retrieval* (IR) problemu, s' tim što se u ovom radu razmatraju samo upitne rečenice kao upiti.

Konkretno, autori porede performanse pretreniranih neuronskih mreža kao što su InferSent, USE-QA u odnosu na BM25, algoritam najčešće korišćen za IR problem. Zadatak modela je pronalazak sličnost između upita i više pasusa, a potom poređaju pasuse u opadajućem redosledu verovatnoće da se u njima nalazi odgovor na pitanje. Korišćeni skupovi podataka su *Stanford Question Answering Dataset* (SQuAD) i Natural Questions (NQ). Kao metrike performansi su korišćene: mean reciprocal rank (MRR) i recall at N (R@N).

Rezultati su pokazali da USE-QA model uspeva da prevaziđe performanse BM25 modela, što znači da se od sada neronske mreže mogu koristiti za pretragu velike količine teksta na semantički način.

C. U trećem radu "*Attention-based Recurrent Neural Networks for Question Answering*" [3] se obrađuje korišćenje "mehanizma pažnje" (engl. *attention*) na vrh rekurentne neuronske mreže. Razmatrana su dva tipa modela: *Match-LSTM* i *Bidirectional Attention Flow* (BiDAF). Ta dva modela kasnije bivaju spojena u jedan ansambl.

Skup podataka koji je korišćen je *Stanford Question Answering Dataset* (SQuAD).

Rešavani problem podrazumeva da se na osnovu pitanja i konteksta koji ide uz dato pitanje predvidi da li u kontekstu postoji početak i kraj odgovora i koji je konkretno odgovor u pitanju.

Mehanizam pažnje je od izuzetnog značaja jer omogućava modelu da se fokusira na delove konteksta koji su najvažniji za pitanje. *Match-LSTM* je korišćen za izračunavanje "pažnje". Na svakoj poziciji konteksta, pažnja se računa reč po reč u *Match-LSTM* ćeliji koja predstavlja omotač oko obične *LSTM* ćelije.

Rezultati koji su ostvareni su prikazani u tabeli 2 gde se može videti da najbolji model ostvaruje F1 meru od 62.8.

	F1	EM	Score
Baseline	48.7	35.2	40.8
Match-LSTM (ours)	58.8	44.6	50.7
Match-LSTM (original)	71.2	61.1	65.8
BiDAF (ours)	62.8	48.6	54.8
BiDAF (original)	77.3	67.7	72.2
Ensemble (ours)	58.4	43.6	49.9

Tabela 2. Prikaz dobijenih rezultata u radu "*Attention-based Recurrent Neural Networks for Question Answering*"

III. Opis skupa podataka

U ovom poglavlju je opisan skup podataka [1] i prikazani su zanimljivi zaključci dobijeni tokom eksplorativne analize podataka.

Originalno, skup podataka je formiran za rešavanje problema *answer sentence selection* i *answer triggering*. Za svaku upitnu rečenicu, dato je nekoliko rečenica u kojima se ne nalazi odgovor na pitanje i manji broj rečenica u kojima se nalazi odgovor na pitanje. Moguć je slučaj da ne postoji rečenica u kojoj se nalazi odgovor na dato pitanje. Zadatak *answer sentence selection* problema je da model od nekoliko ponuđenih rečenica izabere jednu u kojoj se nalazi odgovor na pitanje. U tom slučaju mora postojati bar jedna rečenica u kojoj se nalazi odgovor. U radu je predstavljen i problem *answer triggering* gde je neophodno pre svega odrediti da li u ponuđenim rečenicama postoji rečenica koja sadrži odgovor na pitanje. Za rešavanje našeg problema, svaki par upitna rečenica - izjavna rečenica se posmatra posebno, tj. neophodno je odrediti da li za dati par važi da se u izjavnoj rečenici nalazi odgovor na pitanje ili ne. Nekoliko primera iz skupa podataka su prikazani na slici 1.

index	Question	Sentence	Label
0	how are glacier caves formed?	A partly submerged glacier cave on Perito Moreno Glacier .	0
1	how are glacier caves formed?	The ice facade is approximately 60 m high	0
2	how are glacier caves formed?	Ice formations in the Titlis glacier cave	0
3	how are glacier caves formed?	A glacier cave is a cave formed within the ice of a glacier .	1
4	how are glacier caves formed?	Glacier caves are often called ice caves , but this term is properly used to describe bedrock caves that contain year-round ice.	0

Slika 1. Prvih pet primeraka iz obučavajućeg skupa podataka

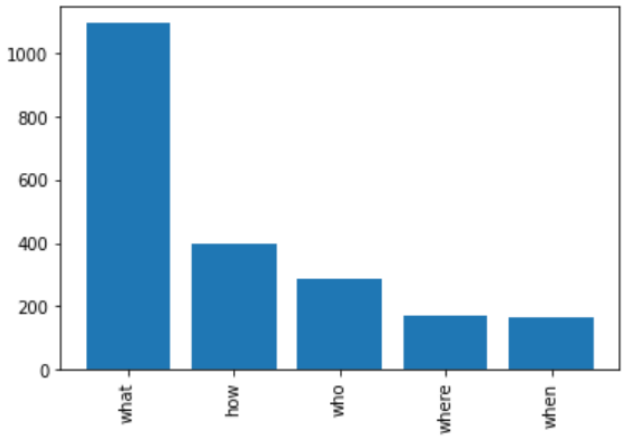
Kao izvor upitnih rečenica su iskorišćeni anonimni upiti upućeni Bing pretraživaču, izvučeni iz logova. Za kandidate rečenica koje potencijalno sadrže odgovor na pitanje je iskorišćen rezime Wikipedia stranice, koji se nalazio među rezultatima pretrage, na koji je korisnik kliknuo. Izjavne rečenice su potom anotirane upotrebom Amazon Mechanical Turk platforme.

Skup podataka je preuzet sa "Microsoft" oficijalnog sajta [4]. Sa pomenutog sajta je moguće preuzeti .zip fajl u kojem je skup podataka već podeljen na trening, validacioni i testni skup u okviru .tsv fajlova. Trening skup sadrži 20355, validacioni skup 2734 a testni skup 6166 primeraka. Svaki primerak je opisan sledećim atributima:

- QuestionID - jedinstveni identifikator pitanja
- Question - tekst pitanja
- DocumentID - jedinstveni identifikator dokumenta koji je korišćen kao izvor rečenica u kojima se potencijalno nalazi odgovor na pitanje
- DocumentTitle - naslov dokumenta
- SentenceID - jedinstveni identifikator izjavne rečenice
- Sentence - izjavna rečenica u kojoj se potencijalno nalazi odgovor na pitanje
- Label - 0 označava da se u datoj izjavnoj rečenici ne nalazi odgovor na pitanje dok 1 označava da se u datoj izjavnoj rečenici nalazi odgovor na pitanje

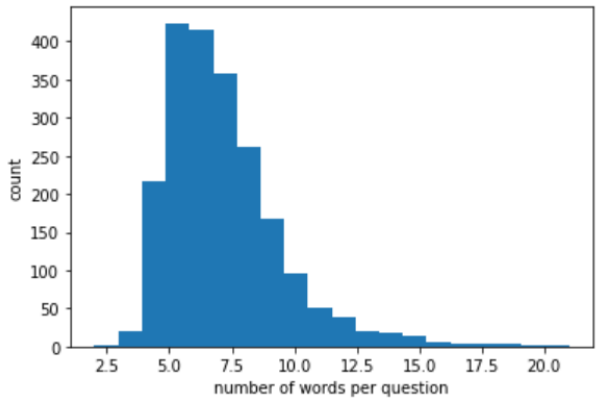
Atributi korišćeni za trening modela su: Question, Sentence i Label.

Među 20355 primeraka u trening skupu, 19315 parova pitanje - izjavna rečenica su označeni sa 0, dok je 1040 parova označeno sa 1, što ukazuje na veliku nebalansiranost među klasama. Jedinstvenih pitanja ima 2118, dok jedinstvenih izjavnih rečenica ima 18816. Najčešće ima 5 do 10 rečenica sa potencijalnim odgovorom po jednom pitanju. Najčešće postoji jedna rečenica sa odgovorom na pitanje među ostalim kandidatima, ali postoje i ređi slučajevi gde ih može biti više (najviše sedam). Na slici 2 je prikazana distribucija prve reči za svako pitanje iz trening skupa.

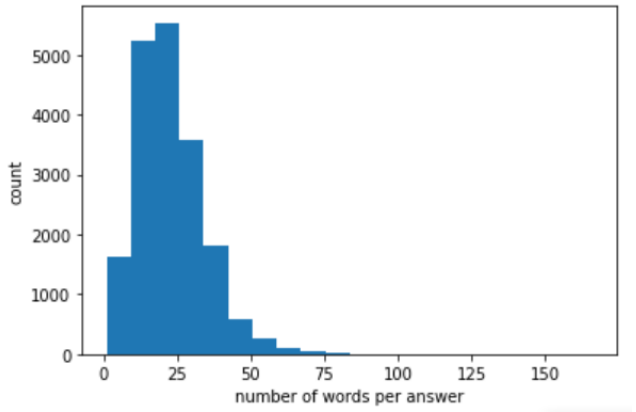


Slika 2. Distribucija prve reči za svako pitanje iz trening skupa

Sa slike 2 se može uočiti da su sva pitanja WH tipa (rešenice koje počinju sa *why/when/what/how* itd..). Na slici 3 je prikazana distribucija broja reči po pitanju, dok je na slici 4 prikazana distribucija reči po rečenicama sa potencijalnim odgovorom.



Slika 3. Distribucija broja reči po pitanju



Slika 4. Distribucija broja reči po rečenicama sa potencijalnim odgovorom

Sa slika 3 i 4 se može zaključiti da je skup podataka formiran pretežno od kraćih rečenica.

Tokom analize trening skupa podataka uočeno je da u originalnom trening .tsv fajlu postoji greška, tj. da nedostaje jedan tab, pa se tokom učitavanja podataka nekoliko parova pitanje - izjavna rečenica pročita kao jedna izjavna rečenica. Problem je ispravljen ručnom izmenom fajla.

Pošto su transformer modeli obučavani nad neobrađenim tekstom, nije primenjeno nikakvo predprocesiranje teksta prilikom obučavanja transformer modela korišćenog u ovom radu, već je tekst takav kakav jeste prosleđen modelu, dok je za CNN i RNN primenjena prethodna obrada podataka kao i lematizacija reči unutar rečenica.

IV. Metodologija

Nakon učitavanja obučavajućeg skupa primenjuju se razne tehnike za obradu teksta. U prvobitnim pristupima se odmah primećuje da je mnogo veći broj netačnih odgovora, s toga se pre svega morala primeniti neka vrsta balansiranja skupa podataka pre samog treniranja. U najvećem broju primera to je postignuto dodavanjem velikih težina u toku obučavanja modela na tačne i veoma malih težina na netačne odgovore.

Da bi se podaci pripremili za ulaz u neuronske mreže koje su trenirane od nule, odrađena je lematizacija, gde je svaka reč unutar rečenica kako pitanja tako i odgovora zamenjena njenom osnovom reči. Na taj način se povećala mogućnost da će se iste reči pojavljivati u više komentara, jer reči koje imaju isti osnovni oblik imaju i isto značenje. Time će se smanjiti vokabular i omogućiti više reči različitog značenja da se nađu u vokabularu. Za lematizaciju je korišćena biblioteka *spacy* [5].

Nakon što su podaci pretprocesirani, spremni su za algoritme mašinskog učenja. Algoritmi i metode koji su korišćeni za rešavanje problema izloženog u radu su:

A. Transformer
B. Rekurentna neuronska mreža
C. Konvolutivna neuronska mreža

A. Nakon rada u kome je objavljen BERT model nastaje procvat u oblasti obrade prirodnog jezika. Ovaj model postavlja nove rekorde u mnogim specifičnim NLP oblastima. Pretreniran je nad ogromnom količinom podataka nad problemom predikcije nedostajućih tokena u rečenici i predikcije naredne rečenice. Ovakav vid treninga je omogućio da mreža nauči semantiku teksta, relacije između reči i gramatiku jezika.

Uobičajen pristup korišćenja pretreniranog transformer modela je prilagođavanje (*fine-tuning*) specifičnom problemu. Prethodni pristupi su zahtevali veliku količinu anotiranih podataka i obučavanje mreže od nule. Sada je potrebno prilagoditi težine modela konkretnom problemu uz pomoć značajno manje anotiranih podataka.

Da bi se ispitalo da li će transformer model imati bolje performanse od ostalih korišćenih modela na problemu koji se rešava u ovom radu isprobani su *all-mpnet-base-v2* [6] i *multi-qa-mpnet-base-dot-v1* [7] transformer modeli iz *sentence-transformers* [8] biblioteke. Osnovu ovih modela čini *Masked and permuted pre-training* (MPNET) [9] model koji je slične arhitekture kao BERT, ali prevazilazi neke od mana BERT modela.

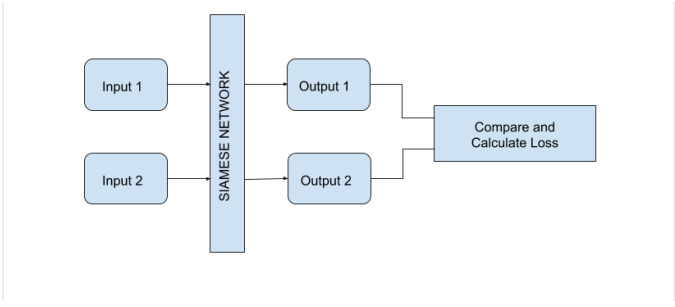
Za treniranje *all-mpnet-base-v2* modela je korišćeno više skupova podataka koji za podatke imaju parove rečenica, najčešće pravljeni za problem semantičke sličnosti. Ukupno je

iskorišćeno milijardu parova rečenica za trening. Model je obučavan tako da na osnovu prve rečenice iz jednog para, prepozna koja je druga rečenica koja pripada tom paru, između još nekoliko nasumično izabranih rečenica. Za ulazni tekst, model proizvodi vektor dužine 768 koji u sebi sadrži semantiku rečenice. Kao metrike sličnosti moguće je koristiti: kosinusnu slučnost, skalarni proizvod (*dot product*) i euklidsku udaljenost. Predviđen je da se koristi kao enkoder za rečenice i kraće paragrafe, za probleme klasterovanja i semantičke sličnosti.

Model *multi-qa-mpnet-base-dot-v1* je treniran nad 215 milliona parova pitanje - odgovor, izvučenih iz više skupova podataka za semantičku pretragu. Tokom treninga, od modela je zahtevano da pronađe paragraf/rečenicu u kojoj se nalazi odgovor na dato pitanje, između više ponuđenih opcija. Za ulazni tekst, bilo to pitanje ili paragraf, model proizvodi vektorsku reprezentaciju dužine 768, nakon čega ih je moguće porediti u vektorskom prostoru. Kao metriku sličnosti moguće je koristiti samo skalarni proizvod. Model je namenjen da se koristi za problem semantičke pretrage.

Za rešavanje našeg problema, ova dva transformer modela su prilagođena trening skupu, najbolji hiperparametri modela su traženi nad validacionim skupom, dok je model krajnje evaluiran nad testnim skupom. Za treniranje modela su isprobane različite funkcije greške: *CosineSimilarityLoss*, *ContrastiveLoss*, *OnlineContrastiveLoss*, *MultipleNegativesRankingLoss* i *TripletLoss*. Kao metrika sličnosti za *all-mpnet-base-v2* model je korišćena kosinusna sličnost a za *multi-qa-mpnet-base-dot-v1* skalarni poizvod.

Model je treniran kao sijamska mreža (*siamese network*), odnosno, i pitanje i rečenica u kojoj se potencijalno nalazi odogovor prolaze kroz istu mrežu, odnosno iste težine se koriste za vektorizaciju i pitanja i rečenice u kojoj se potencijalno nalazi odgovor. Kasnije se te težine menjaju u odnosu na funkciju greške i metriku sličnosti. Vizuelni prikaz se nalazi na slici 5. Ovaj pristup omogućava da se pitanje i rečenica u kojoj se potencijalno nalazi odgovor mogu vektorizovati zasebno, nezavisno jedno od drugog, što omogućava da se oni vektorizuju samo jednom, a kasnije da se koristi određena metrika sličnosti kako bi se odredila sličnost među njima. To ima primenu u uslovima velike količine podataka, tj. poređenja velike količine vektora.



Slika 5. Sijamska mreža

Tokom treniranja modela korišćena je ručna implementacija *early stopping*-a. Pre treniranja modela izračuna se makro f1 mera nad validacionim skupom. Nakon svake epohe se ponovo računa makro f1 mera i upoređuje se sa prethodnom. Ukoliko je trenutna makro f1 mera manja od prethodne, treniranje se zaustavlja i model iz prethodne epohe se sačuva na disk za kasniju evaluaciju. Najčešće je bilo potrebno jedna do tri epohe kako bi model dostigao maksimalnu makro f1 meru.

Nakon svake epohe, da bi se odredila najbolja makro f1 mera, isprobano je više različitih *threshold*-a. U slučaju kosinusne sličnosti, *threshold* koji daje najbolju makro f1 meru na validacionom skupu je tražen u opsegu od 0 do 1 sa korakom od 0.01, dok je u slučaju skalarnog proizvoda taj opseg bio od 0 do 100 sa korakom 1. Ukoliko je sličnost između pitanja i izjavne rečenice iznad definisanog *threshold*-a, tada se smatra da ta rečenica sadrži odgovor na pitanje, ukoliko je sličnost ispod definisanog *threshold*-a, tada se smatra da rečenica ne sadrži odgovor na pitanje.

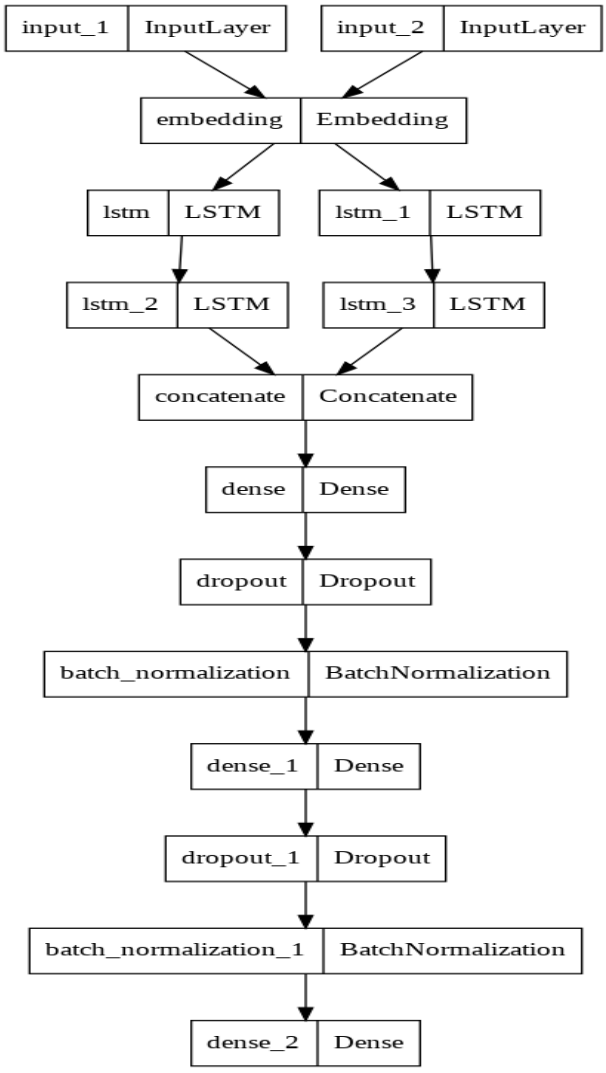
Hiperparametri koji su bili isti za treniranje svih varijacija transformer modela su:

- *train batch size* = 16
- warmup steps = proizvod veličine trening skupa, inicijalnog broja epoha (ne računajući *early stopping*) i faktora skaliranja (koji je bio 0.1)

B. Rekurentne neuronske mreže ne uče isključivo reprezentaciju za svaku reč, nego i reprezentaciju cele rečenice. Korisne su jer čuvaju informaciju o trenutnom obeležju, kao i susedna obeležja za predviđanje, ali nekad ne uspeju da uhvate susednu reprezentaciju drugih sekvenci. LSTM (*Long short-term memory*) može pomoći u rešavanju ovog problema jer razume kontekst zajedno sa nedavnom zavisnošću. Otuda su LSTM posebna vrsta rekuretnih neuronskih mreža gde razumevanje konteksta može biti korisno.

Ulazi za parove pitanje - odgovor su odvojeno konvertovani u *embedding* matrice. Nakon čega su korišćena dva sloja LSTM-a i za pitanja i za odgovore gde oba sloja koriste po 50 neurona, uz to da prvi daje celu povratnu sekvencu kao ulaz u sledeći RNN sloj, postavljanjem atributa *return_sequences* na *True*. Potom su LSTM slojevi za pitanja i odgovore spojeni u jedan, radi klasifikacije. Zatim je dodat *Dense* sloj od 200 neurona sa *Rectified Linear Unit* (ReLu) aktivacionom funkcijom, *Dropout*-om od 0.2 i primenjen je *BatchNormalization* kako bismo sprečili pretreniranost (engl. *Overfitting*). Ovaj pristup korišćenjem *Dense*, *Dropout* i *BatchNormalization*-a je ponovljen naknadno još jednom i na kraju kao izlaz mreže se nalazi *Dense* sloj sa *sigmoid* aktivacionom funkciom. *Loss* funkcija za koju smo se opredelili jeste *binary_crossentropy* uz *adam* optimizer sa korakom učenja od 0.001.

Jedan od vrlo važnih atributa koji se koristio pri treniranju mreže jeste i *class_weights* koji je bilo potrebno primeniti radi pristrasnosti mreže ka netačnim odgovorima označenim sa 0 usled nebalansiranosti samog skupa podataka.



Slika 6. Arhitektura rekurentne neuronske mreže

Kao nadogranja na postojeću rekurentnu mrežu, za inicijalizaciju težina vektora reči korišćen je pretrenirani *GloVe 6B embedding*, a model je isproban sa 50, 100, 200 i 300 dimenzija vektora reči.

GloVe se ne oslanja samo na lokalne informacije o kontekstu reči već gleda globalnu statistiku pojavljivanja reči za dobijanje vektora reči.

Hiperparametri koji su korišćeni pri dobijanju najboljih rezultata na modelu rekurentne mreže su:

- *train batch size* = 32
- class_weights = 1:8
- epochs = 5

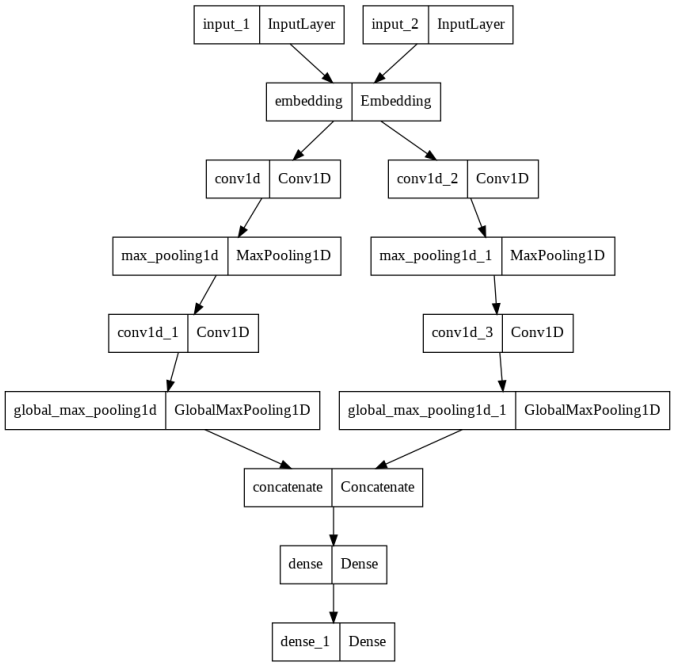
C. Konvolutivne neuronske mreže se baziraju na klasifikaciji ulaznih podataka, tako da izlaz može biti jedna ili više klasa. Ideja konvolutivnih mreža je da se postavi veći broj slojeva za otkrivanje bitnih karakteristika ulaznih podataka.

Ulaz u mrežu su dve sekvence 256-dimenzionalnih vektora, gde prvi reprezentuje pitanje a drugi odgovor. Mreža uči vektor reprezentacije za vokabular reči koji uključuje čitav skup tokenizovanih reči.

Kod konvolutivne mreže imamo konvolutivni sloj, sloj sažimanja i potpuno povezane slojeve. Veze između ovih slojeva su aktivacione funkcije, u ovom slučaju to su *ReLu* i *Softmax*. Sažimanje maksimumom (engl. *MaxPooling*) je važan koncept konvolutivnih neuronskih mreža jer se uklanjanjem vrednosti, koje nisu maksimalne, smanjuje izračunavanje. *Pooling* sloj se obično dodaje nakon konvolucionog sloja uglavnom radi smanjenja dimenzionalnosti mape karakteristika radi računske efikasnosti, što kasnije može poboljšati stvarne performanse.

Korišćene su pretrenirane *GloVe 6B* reprezentacije vektora za *embedding* koji pruža univerzitet Stanford.

Veliku ulogu pored kategorizacije izlaznih rezultata imalo je i rano zaustavljanje koje sprečava da se model ne *overfit*-uje već nakon nekoliko iteracija treniranja.



Slika 7. Arhitektura konvolutivne neuronske mreže

V. Rezultati

Svi modeli mašinskog učenja su trenirani nad istim trening skupom i testirani nad istim test skupom podataka. Način na koji su podaci dobavljeni i procesirani je pomenut u poglavlju III. Kao metrika za upoređivanje korišćena je makro F1 mera (engl. *macro* F1 *measure*) kako bismo dobili realističnije rezultate zbog velike nebalansiranosti među labelama. Makro F1 mera drastično smanjuje krajnji rezultat ukoliko model ne radi dobro za manjinsku klasu. Da bi mogli da objasnimo F1 meru, potrebno je prethodno objasniti preciznost (engl. *Precision*), tačnost (engl. *Accuracy*) kao i odziv (engl. *Recall*). Preciznost predstavlja udeo dobro predviđenih primera određene klase u ukupnom broju primera koje je model svrstao u datu klasu. Time dobijamo meru koji deo rezultata u jednoj klasi je uspešno klasifikovan. Odziv predstavlja osetljivost modela, odnosno, pokazuje koliko je relevantnih rezultata algoritam vratio.

Odziv se računa po formuli:

$$Recall = \frac{True\ positives}{True\ Positives + False\ Negatives}$$

Tačnost se računa po formuli:

$$Accuracy = \frac{True\ positives + True\ Negatives}{All\ Samples}$$

Preciznost se računa po formuli:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

F1 mera kombinuje preciznost i odziv i računa se po sledećoj formuli:

$$\text{F1 Score} = \frac{2\ x\ (Precision\ x\ Recall)}{Pre}$$

Makro F1 mera predstavlja srednju vrednost F1 mera od svake klase posebno.

Konačni rezultati nad testnim skupom podataka za sve korišćene modele u ovom radu su dati u tabeli 3.

Model	Konfiguracija	Makro F1
CNN	broj epoha=15	0.5117
CNN	GLOVE embeddings broj epoha = 15	0.5821
RNN	broj epoha=7	0.5269
RNN	GLOVE embeddings broj epoha=7	0.5441
multi-qa-mpnet-base-dot-v1	broj epoha=0 *threshold*=28	0.6819
all-mpnet-base-v2	broj epoha=0 *threshold*=0.7	0.6747
all-mpnet-base-v2	broj epoha=2 *threshold*=0.38 *loss=CosineSimilarityLoss*	**0.7378**
all-mpnet-base-v2	broj epoha=2 *threshold*=0.67 *loss=TripletLoss*	0.735

Tabela 3. Rezultati nad testnim skupom za sve modele korišćene u radu

Iz tabele se može videti da najbolji rezultat postiže *all-mpnet-base-v2* model sa *CosineSimilarityLoss* funkcijom greške. Približne rezultate postiže *all-mpnet-base-v2* model sa *TripletLoss* funkcijom greške. Ukoliko se u koloni Konfiguracija u tabeli 1 nađe broj epoha=0 to znači da je taj model iskorišćen takav kakav jeste, bez dodatnog treniranja (engl. *out-of-the-box*).

Za *MultipleNegativesRankingLoss* funkciju greške, makro f1 mera nad validacionim skupom je bila ispod 0.7, pa ta funkcija greške nije više razmatrana. Makro f1 mera na validacionom skupu za *ContrastiveLoss* i *OnlineContrastiveLoss* funkcije greške je bila iznad 0.7, međutim nad testnim skupom je bila ispod 0.7. Za gotovo sve funkcije greške *multi-qa-mpnet-base-dot-v1* model se pokazao blago lošije u odnosu na *all-mpnet-base-v2* model. Moguć razlog za to je što je *all-mpnet-base-v2* model dotreniran nad većom količinom podataka. Drugi moguć razlog može ležati u tome što su tekstovi koji potencijalno sadrže odgovore na pitanja u ovom skupu podataka kraći u odnosu na tekstove nad kojim je treniran *multi-qa-mpnet-base-dot-v1.*

Na slici 8 je prikazan klasifikacioni izveštaj (engl. *classification report*) gde su prikazane preciznost, tačnost i F1 mera za svaku labelu posebno. Izveštaj je dobijen na osnovu modela koji se najbolje pokazao u ovom radu.

	precision	recall	f1-score	support
0	0.98	0.97	0.97	5868
1	0.49	0.51	0.50	293
accuracy			0.95	6161
macro avg	0.73	0.74	0.74	6161
weighted avg	0.95	0.95	0.95	6161

Slika 8. Klasifikacioni izveštaj

Posmatrajući sliku 8, može se uočiti da su metrike za labelu 0 značajno bolje u odnosu na labelu 1. To je i razumljivo, uzimajući u obzir veliku nebalansiranost skupa podataka. Težini problema takođe doprinosi vrlo mali broj primeraka u trening skupu sa labelom 1 (oko hiljadu), pa modeli nisu imali dovoljno podataka kako bi se prilagodili problemu.

Primenom *GloVe 6B* pretrenirane reprezentacije se ostvaruje 1-5% bolji rezultat u odnosu na osnovnu verziju CNN i RNN mreže.

Kako bi rezultati bili što merodavniji, uzet je prosek od 5 obučavanja nad istom arhitekturom i parametrima za RNN i CNN modele po sledećoj formuli:

$$\text{Result} = \frac{R1 + R2 + R3 + R4 + R5 + \dots RN}{N}$$

Rezultati pokazuju da je za ovaj problem korisnije koristiti transfer learning nego trenirati mrežu od nule. Moguće je da je transformer model u svojim težinama sačuvao mnogo više informacija nego što su to mogle da nauče RNN i CNN mreže tokom treniranja. Druga mogućnost zašto transformer postiže značajno bolje rezultate od ostalih modela je to što je on dosta kompleksniji model i bolje "razume" semantiku teksta što su pokazali dosadašnji rezultati u drugim radovima.

VII. Zakljucak

U ovom radu su prikazane različite tehnike rešavanja problema predikcije sličnosti pitanja i odgovora u sklopu QA problema. Skup podataka se sastojao od velikog broja rečenica od kojih su jedne bile pitanja a druge odgovori. Korišćeni atributi su pitanje, odgovor i labela sa tačnošću datog para. Na početku je izvršena eksplorativna analiza, izbačene su stop reči, izvršena je lematizacija. Posle pretprocesinga skup podataka je bio spreman da se podvrgne određenom skupu algoritama, koji su vršili predikciju sličnosti na osnovu teksta pitanja i dogovora. Pre ovoga je skup podataka podeljen na obučavajuće podatke, validacione, kao i test podatke.

Jedna od glavnih razlika ovog rada sa ostalim radovima koji se bave rešavanjem QA problema je taj što se većina radova svodi na to da unutar skupa podataka postoji grupa u kojoj se nalazi nekoliko parova sa istim pitanjem i različitim odgovorima gde je najčešće jedan ili ni jedan od njih tačan. Samim tim model može bolje da nauči kako izgledaju netačni odgovori na isto pitanje a kako tačan odgovor. U radu koji je ovde opisan ne postoje grupacije i svaki par se uči zasebno, samim tim je problem značajno teži za rešavanje.

Za predikciju sličnosti pitanja i odgovora korišćeno je nekoliko algoritama. Za upoređivanje je korišćena makro F1 mera. Korišćenjem ove mere može se zaključiti kakve rezultate daju pojedini algoritmi na skupu podataka čije su labele nebalansirane. Dobri rezultati su dobijeni korišćenjem rekturentne i konvolutivne neurosnke mreže, dok su još bolji rezultati dobijeni upotrebom transformera, što se moglo očekivati jer je transformer značajno kompleksniji model od prethodna dva.

Kao poboljšanja upotrebljenih algoritama, upotrebljena je pretrenirana *GloVe* reprezentacija kao ulazni vektor u CNN i RNN.

Dalje unapređenje performansi bi se moglo postići pravljenjem grupacija u odnosu na parove pitanja i odgovora gde bi model naučio kako izgledaju tačni a kako netačni odgovori za dato pitanje ali bi se tada ovaj rad odaljio od svog cilja. Dodavanje nekoliko *Dense* slojeva na izlaz iz transformer mreže bi moglo pozitivno uticati na rezultat. Takođe, moguće poboljšanje bi se moglo postići augmentacijom trening skupa u cilju povećanja parova sa tačnim odgovorom, radi postizanja bolje balansiranosti.

Literatura

[1] WikiQA: A Challenge Dataset for Open-Domain Question Answering https://aclanthology.org/D15-1237/

[2] ReQA: An Evaluation for End-to-End Answer Retrieval Models https://arxiv.org/pdf/1907.04780.pdf

[3] Attention-based Recurrent Neural Networks for Question Answering https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761224.pdf

[4] WikiQA corpus dataset https://www.microsoft.com/en-us/download/details.aspx?id=52419

[5] Spacy biblioteka https://spacy.io/

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[7] https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1

[8] Sentence-transformers repozitorijum https://github.com/UKPLab/sentence-transformers

[9] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. Advances in Neural Information Processing Systems, 33, 16857-16867.