Preporuka vesti uzimajući u obzir da su vesti istinite

Luka Kričković

Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Trg Dositeja Obradovića 6

21000 Novi Sad

e-pošta: krickovic.e220.2021@uns.ac.rs

Đorđije Kundačina

Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Trg Dositeja Obradovića 6

21000 Novi Sad

e-pošta: kundacina.e276.2021@uns.ac.rs

*Apstrakt*—U ovom radu opisan je sistem koji za glavni entitet ima vijest (članak) i sastoji se iz dva podsistema. Prvi podsistem zadužen je za klasifikaciju vijesti na istinite i lažne vijesti. Dok drugi dio sistema za zadatak ima da analizira aktivnosti korisnika i da na osnovu toga, preporuci korisniku slične vesti (članke). Klasifikacija vijesti, odnosno članaka uradjena je uz pomoc vise metodologija i rezultati svih metodologija međusobno su upoređeni, dok je preporuka vesti (članaka) urađena uz pomoć LSTUR (**L**ong **S**hort **T**erm **U**ser **R**epresentations) sistema. Ideja rada proistekla je iz potrebe da se lažne vesti odvoje od “istinitih”, kao i da korisnici jednostavnije i brže dobijaju njima interesantne vesti, na osnovu prethodnih interesovanja.

Ključne reči—Fake or real news, classification, system recommendation, lstm, bert, tf idf, glove, lstur

# Uvod

Internet portali su postali defakto način konzumiranja vesti većini ljudi koji imaju pristup potrebnim tehnologijama. Zbog velike količine tekstova koji se objavljuju svaki dan, ljudi su ograničeni samo izvorima na koje su navikli, poput omiljenih portala i slično. Pretrage ključnih reči tema na Google i na društvenim mrežama često dovodi do promovisanja malicioznogsadržaja, zbog politike kompanija da se prioritet daje sponzorisanim rezultatima. Kako bi se borilo protiv ovoga, postoje platforme koje skupljaju vesti sa različitih izvora, pa ih preporučuju korisnicima i koriste svoje sisteme za filtriranje nepodesnih članaka. Primeri ovakvih sistema su “Microsoft News” i “Google News”. Pošto su ovi sistemi u vlasništvu kompanija koje ih koriste, tehnologije koje napajaju ove sisteme nisu bile dostupne do nedavno, kada su Microsoft objavili “MIND” [1], u kome detaljno opisuju postojeća rešenja. Kao rezultat ovog rada, sistemi za preporuku vesti su postali javno dostupni i niz novih tehnologija je razvijen da sve bolje i bolje uče šablone pregleda pojedinačnih korisnika. Problem koji se često javlja kod ovih radova jeste što postojeći modeli često nemaju mehanizme za sprečavanje malicioznih članaka, nego se filtriranje vrši u fazi obrade podataka, ili se ne vrši uopšte.

Cilj ovog projekta je kreiranje sistema koji se sastoji iz dve celine: podsistem za preporuku vesti i podsistem za filtriranje malicioznih članaka. Za rešavanje problema preporuke članaka upotrebljen je MIND (eng. “**Mi**crosoft **N**ews **D**ataset”) skup podataka [1], koji sadrži podatke o istorijatu klikova korisnika, tekstove članaka, kao i metapodatke poput naslova, kategorije i podkategorije teksta i slično. Tekstovi se provlače kroz enkoder članaka, što proizvodi embedding vektor za svaki pojedinačni članak, dok se na osnovu istorijata klikova korisnika pravi embedding vektor korisnika. Ova dva vektora se spajaju na specifičan način koji ćemo detaljno objasniti u daljem tekstu, da bi se kao izlaz iz modela dobio vektor identifikatora članaka koji su preporučeni za pojedinačnog korisnika. Podsistem za filtriranje malicioznih članaka se tretira kao nezavisni sistem, pošto se trenira na nezavisnom skupu podataka zbog manjka malicioznih članaka u očišćenom MIND skupu podataka. Za ovaj podsistem smo prvenstveno iskoristili “Fake and real news dataset”, od autora Clément Bisaillon [7], koji sadrži tekstove, naslove i teme članaka. Na osnovu ovog skupa, kao i radova koje ćemo opisati u narednom poglavlju, napravili smo eksperimente sa LSTM, BERT, GloVe i TF-IDF modelima.

U narednom poglavlju će biti opisani radovi relevantni za naše probleme, kao i njihove metodologije i rezultati. Poglavlje tri opisuje skupove podataka koje smo upotrebili, kao i procese kroz koje smo izmenili originalne skupove da bismo dobili što bolje rezultate. Metodologija za rešavanje problema preporuke vesti i filtriranje malicioznih članaka je detaljno opisana u poglavlju 4. Nakon toga su predstavljeni rezultati naših eksperimenata i na kraju rada je izveden zaključak.

# Slična istraživanja

Wu et al. predstavljaju prvi javno dostupni skup podataka sa velikim brojem članaka i utiscima korisnika MSNews platforme, uz šta i detaljno opisuju postojeće tehnologije za preporuku vesti [1]. Ovaj rad je bio presudan za naš projekat, jer nudi sumarizaciju postojećih modela i daje rezultate i preporuke za dalje optimizacije. Iz ovog rada smo pronašli metode koje smo dalje kombinovali i iskoristili u implementaciji sistema za preporuku vesti.

         Maramreddy et al. u radu [2] dele arhitekturu njihovog sistema za preporuku vesti, koji se, za razliku od našeg sistema, većinski fokusira na načine prikupljanja podataka o korisnicima, što se realizuje čitanjem istorije pretrage korisnika i analizom kolačića na korisnikovom pretraživaču. Informacije koje smo uspeli da primenimo iz ovog rada su podaci o obradi ulaznih tekstova. Autori predlažu da se tekstovi članaka tokenizuju, da se svi tokeni konvertuju u mala slova, izbacuju stop reči, lematizuju i da im se izvlači koren (eng. Stemming). Nakon ove obrade, poredili su par popularnih pristupa, konkretno Bag of words, TF-IDF i Word2Vec kako bi prezentovali prednosti i mane svakog pristupa. Na osnovu ovoga smo u sistemu za detekciju malicioznih članaka izvršili eksperiment sa predloženim TF-IDF pristupom i Multinomial Naïve Bayes klasifikatorom i postigli odlične rezultate, o čemu će biti reči u daljem tekstu.

         Wu et al. su detaljno opisali NPA (eng. Neural **N**ews Recommendation with **P**ersonalized **A**ttention) sistem za preporuku [3], čiji smo enkoder vesti implementirali u okviru našeg rešenja. Ovaj rad je značajan, jer predlaže upotrebu personalizovanog enkodera vesti, umesto generičkog kojeg predlažu alternativni sistemi. Enkoder vesti se sastoji iz dropout sloja, konvolucionog sloja, i personalizovanog attention mehanizma, o kome će detaljno biti reči u metodologiji.

         Mingxiao An et al. u radu [4] predstavljaju alternativu prethodno spomenutom NPA sistemu za preporuku, takozvani LSTUR (**L**ong **S**hort **T**erm **U**ser **R**epresentations) sistem. Na osnovu ovog modela smo implementirali enkoder korisnika, jer se u predikcije ubacuje podatak o datumu pregleda odabranog članka. Ovime se obezbeđuje da se dugoročni interesi korisnika uvek preporučuju, dok se kratkoročni interesi preporučuju samo u odgovarajućem vremenskom intervalu, odnosno ukoliko korisnik pregleda vest o nečemu što mu verovatno predstavlja kratkoročni interes, sistem će mu preporučivati slične članke samo u neposrednoj budućnosti. Detalji o arhitekturi ovog modela će biti opisani u metodologiji.

U radu [5], Winklemann et al. opisuju rešenje za filtriranje lažnih vesti (eng. “fake news” ). U ovom radu je opisan način prikupljanja podataka za kategorizaciju, kao i specifičnosti koje smo koristili u našim eksperimentima, poput metoda pretprocesovanja podataka, kao i upotrebe tf-idf i lstm modela za klasifikaciju tekstova članaka na redovne i lažne (eng. real, fake).

# Opis skupova podataka

Pošto se naš sistem sastoji iz dve celine, upotrebljena su dva skupa podataka za treniranje svakog podsistema pojedinačno. Podsistem za preporuku vesti koristi MIND (eng. **Mi**crosoft **N**ews **D**ataset) [6] skup podataka, dok se podsistem za detekciju malicioznih vesti koristi Fake News skup podataka [7].

         Mind skup podataka se sastoji iz 160.000 članaka na engleskom jeziku, kao i 15 miliona utisaka od milion korisnika MSNews platforme. Sastoji se iz četiri datoteke, „behaviours.tsv“, „news.tsv“, „entityembedding.vec“ i „relationembedding.vec“. Behaviours datoteka sadrži utiske korisnika, modelovane kroz identifikatore utiska i korisnika čiji je utisak, vremena kreiranja utiska, istorijata članaka koje je korisnik pregledao, skupa naslova koji su predstavljeni korisniku anotirani sa 1 ili 0 u zavisnosti od toga da li je korisnik kliknuo na naslov ili ne. News dokument sadrži podatke o člancima poput identifikatora članka, kategorije, podkategorije, naslova, apstrakta, linka ka članku, kao i podatke poput embedding-a naslova i apstrakta. Entity i relation embedding datoteke sadrže embedding vektore entiteta i relacija između entiteta. Za potrebe ovog projekta smo koristili demo verziju ovog skupa, kako bismo istrenirali model u prihvatljivom vremenskom periodu. Ovaj skup je prerađen samo tako što je vektorizovan upotrebom word2vec algoritma. Ovaj skup je podeljen na trening, test i validacioni podskup. Zbog postojanja microsoft-ovog recommenders API-a, nisu ručno preuzimani skupovi i deljeni u podskupove, nego su pozivom metode  „download\_deeprec\_resources“ preuzeti „MINDdemo\_train“ i „MINDdemo\_dev“, od kojih prvi služi za trening i test, a drugi za validaciju.

         Fake news skup podataka se sastoji iz 17903 jedinstvena članka. Svaki članak je opisan pomoću naslova, teksta, teme i datuma kada je objavljen. Iako teme ne ulaze u proces klasifikacije u našim modelima, mogu predstavljati indikator potencijalnog disbalansa u našem skupu. Nakon dubljeg istraživanja nad skupom podataka ustanovili smo da postoji 39% tema označenih sa „news“ i 29% označenih sa „politics“. Kada se ovaj skup razdeli na istinite i lažne članke, disbalans postaje sve naglašeniji, jer se u istinitim člancima pojavljuje samo dve teme. Kako bismo rešili ovaj problem, dodali smo redove iz još dva skupa, „All the news“ [8] i „Fake news“ [9] na način koji ćemo opisati u metodologiji. Ovaj skup podataka je potom obrađen tako što su izbačeni nepotrebni karakteri poput uglastih zagrada, zareza, dvotačka i slično. Potom su izbačeni brojevi i stop reči. Ovako obrađen skup podataka je poslat u sve eksperimente sa klasifikacijom malicioznih vesti. Ovaj skup je podeljen na 80% za trening i 20% za test uz pomoć sklearn „train\_test\_split“ metode. Za ovakvu podelu smo se odlučili na osnovu preporuka iz rada [7].

# Metodologija

Metodologija ovog rešenja se može podeliti u dve veće celine: podsistem za preporuku vesti i podsistem za filtriranje malicioznih članaka. Podsistem za preporuku vesti podrazumeva kreiranje modela i optimizaciju hiper parametara. Podsistem za filtriranje članaka se sastoji iz pripreme skupova podataka i četiri eksperimenta: LSTM, TF-IDF, Bert i GloVe klasifikacije tekstova.

1. Podsistem za preporuku vesti

Ovaj podsistem se sastoji iz enkodera korisnika i enkodera vesti. Obe komponente funkcionišu tako što dobijaju niz ulaznih podataka i kao rezultat proizvode dva vektora, koji zajedno daju predikciju. Prikaz arhitekture ovog dela sistema je dat na slici 1.

A picture containing box and whisker chart

Description automatically generated

Slika 1. Prikaz arhitekture enkodera vesti

Prvi korak u enkoderu vesti jeste pretvaranje svake pojedinačne reči iz naslova članka u vektor upotrebom word2vec algoritma. Reči su na slici naznačene sa [w1, w2, …, wm], gde je m ukupan broj reči u naslovu, dok su vektori označeni sa [e1, e2, …, em]. Sledeći korak je konvolucioni sloj, koji služi za izvlačenje konteksta iz niza reči. Primer koji je dat u radu [3] je naslov „best Fiesta bowl moments“, gde ovaj sloj dodeljuje najveću vrednost „Fiesta bowl“ delu rečenice. Dobijeni vektori se prosleđuju u attention mrežu, koja funkcioniše na nivou reči. Ono što je jedinstveno kod ovog pristupa, u odnosu na druga rešenja, je to što ova attention mreža uključuje i podatak o enkodiranom identifikatoru korisnika za koga se vrši predikcija. Ovo postižemo slanjem korisnikovog identifikatora u potpuno povezani sloj, nakon čega dobijene podatke spajamo sa vektorizovanim rečima dobijenim iz konvolucionog sloja upotrebom dot sloja.

Specifičnost naše komponente za enkodovanje korisnika u odnosu na druge metode jeste to što modeluje korisnikove kratkotrajne i dugotrajne interese, što znači da ukoliko pretpostavi za neki članak da je kratkotrajni interes korisnika (na primer naslov nekog filma ili serije), preporučivaće mu slične članke samo u neposrednoj budućnosti. Izgled arhitekture ovog modela je prikazan na slici 2.

Diagram

Description automatically generated

Slika 2. Prikaz arhitekture enkodera korisnika

Prvo se svaki kandidat članka provlači kroz enkoder vesti, nakon čega se svaki članak iz istorijata pregleda korisnika provlači kroz GRU (eng. Gated Recurrent network). Embedding identifikatora našeg korisnika služi za inicijalizaciju skrivenog stanja GRU. Izlaz iz ove mreže predstavlja vektor koji se u dot proizvodu spaja sa embeddingom kandidatskih članaka. Upotrebljene hiper parametre smo dobili na osnovu više stavki: vrednosti koje recommenders biblioteka nalaže, parametara dobijenih empirijski, odnosno kroz više iteracija treniranja modela (learning rate: 0.0001, optimizer: adam), kao i na osnovu radova [3] (veličina prozora konvolucionog sloja: 3, broj filtera: 400, attention hidden dim: 200) i [4] (broj GRU slojeva: 400, loss funkcija: cross entropy loss).

B. Priprema fake news skupa podataka

Kada smo uvezli „Fake and Real news dataset” [7], prvi korak je bio da analiziramo zastupljenost tema članaka, jer ukoliko postoji prevelik broj članaka iz jedne pojedinačne teme, znači da će svi modeli da rade bolje za tu temu nego za ostale. Ono što smo primetili jeste da su vesti i politika dominantne teme, sa preko 50%. Grafik sa prikazom svih tema je dat na slici 3.

Chart, pie chart

Description automatically generated

Slika 3. Grafički prikaz svih tema zastupljenih u neobrađenom „Fake and Real news dataset“.

Kako bismo doprineli raznolikosti ovog skupa, prvo dodajemo još jedan skup lažnih vesti, konkretno „Fake news“ [9]. Pošto ovaj skup podataka sadrži i istinite i lažne članke, prvi korak je da filtriramo samo lažne članke, nakon čega skup redukujemo samo na potrebne kolone: naslov, tekst, tema, istinit ili lažan. Nakon ovoga, kada smo ovaj skup dodali na postojeći, dobili smo raspodelu prikazanu u tabeli 1.

|  |  |
| --- | --- |
| News | 0.365288 |
| Politics | 0. 276125 |
| left-news | 0.179980 |
| Government News | 0.063370 |
| US\_News | 0.031604 |
| Middle-east | 0.031403 |
| bs | 0.024258 |
| conspiracy | 0.017356 |
| satire | 0.005893 |
| junksci | 0.004117 |
| fake | 0.000605 |

Tabela 1. Nova, normalizovana raspodela tema za lažne članke (raspon od 0 do 1, uključujući i 0 i 1).

Sledeći korak je bio dodavanje „All the news“ [8] skupa podataka. Problem sa ovim skupom je bio nedostatak kolone sa temama svakog pojedinačnog članka, pa kako bismo znali otprilike koliko je opširan ovaj skup, upotrebili smo LDA (Latent Dirichlet Allocation) za ekstrakciju teme. Kako bismo upotrebili ovaj model, prvo smo morali da tokenizujemo i lematizujemo tekst, pa da svaku reč pretvorimo u korenski oblik (eng. stemming), jer se LDA vektorizator bazira na broju pojava svake reči, pa bi različiti oblici iste reči mogli dovesti do grešaka u estimaciji. Nakon prerade, vektorizovali smo tekst upotrebom CountVectorizer klase iz sklearn biblioteke. Prilikom vektorizacije su izbačene i stop reči, tekst je konvertovan u mala slova i postavljen je filter da svaka reč ima barem tri karaktera i da maksimalno može predstavljati 10% teksta. Potom je vektor poslat na LDA model, čiji su hiper parametri određeni empirijski i na osnovu članka [10]. Nakon ovoga, najpopularnijih deset tema kojih smo dobili su: film, health, game, gun, immigration, market, street, student, car i sander. Možemo da primetimo da su gun i immigration političke teme, ali ostatak je prilično raznovrstan, pa možemo pretpostaviti da se to odnosi i na ostatak skupa.

Ubacujemo ovaj skup u početni skup podataka tako što izdvajamo istinite članke, od kojih potom nasumično izbacujemo 50%. Iz novog skupa potom izbacujemo n članaka, tako da je ukupan broj istinitih članaka na kraju jednak broju lažnih. Formula za računanje n je:

n = new\_true.shape[0] + true\_subset.shape[0] - total\_true\_count, gde new\_true predstavlja novi skup podataka kojeg integrišemo u stari, true\_subset predstavlja podskup od 50% starih istinitih članaka i total\_true\_count predstavlja ukupan broj članaka što nam treba, odnosno ukupan broj lažnih članaka.

C. Klasifikacija uz pomoć LSTM

Ono što svi eksperimenti imaju zajedničko je pretprocesovanje. Konkretno, iz svih tekstova se izbacuju stop reči, kao i nebitni karakteri poput uglastih zagrada i slično. Nakon toga, ovi tekstovi se tokenizuju i dodaje im se padding, kako bi se garantovalo da je svaki token iste dužine. Pošto imamo bool vrednosti za klase (fake može biti True ili False), moramo ih konvertovati u numerički oblik, što radimo uz pomoć get\_dummies metode pandas biblioteke. Ovako se True mapira na 1, a False na 0. Potom se skup podataka deli na trening i test podskup u raspodeli 80:20, na osnovu članka [11].

Naš klasifikator treba prvo da pretvori tokene u embedding vektore, nakon čega koristi lstm za klasifikaciju i poslednji sloj je potpuno povezan sa softmax aktivacionom funkcijom. Između embedding i lstm sloja se dodaje dropout sloj, kako bi se izbeglo pretreniranje (eng. overfitting). Hiper parametre za ovaj model smo takođe preuzeli iz članka [11], uz izmenjen batch\_size, kako bismo treniranje prilagodili vektorima većih dimenzija.

D.   Klasifikacija uz pomoć TF-IDF i Naive Bayes

Kao i za prethodni eksperiment, tekstovi svih članaka su prvo očišćeni, iako za TF-IDF ovo ne bi trebalo da pravi preveliku razliku zbog IDF dela algoritma. Međutim, kada smo uporedili skup sa i bez čišćenja, ispostavilo se da se nad očišćenim skupom dobija veća preciznost u odnosu na neočišćen, ali za ±2 %. Nakon čišćenja, tekstovi se šalju u TfidfVectorizer klasu i bool klase se pretvaraju u numeričke oznake kao u prethodnom eksperimentu. Nakon što se tekstovi vektorizuju, skup se razdvaja na trening i test u razmeri 80:20 kao i u prethodnom eksperimentu, takođe na osnovu članka [11]. Potom se vektori klasifikuju uz pomoć Multinomial Naïve Bayes klasifikatora.

E. Klasifikacija uz pomoć GloVe i sekvencijalne mreže

Zbog prirode GloVe vektorizatora, obrada teksta koju smo koristili u prethodna dva eksperimenta dolazi do drastičnijeg izražaja. Nakon čišćenja, tekst se tokenizuje istim tokenizatorom kojeg smo upotrebili u LSTM eksperimentu. Tokene ponovo padujemo, kako bismo garantovali da su svi iste dužine. Da bi se kreirao sekvencijalni model, potrebno je kreirati embedding matricu od GloVe i dobijenih tokena. Ovo se postiže kreiranjem dvodimenzionalnog vektora popunjenog nulama. Nakon toga se vrednosti svakog reda popunjavaju GloVe vektorskim reprezentacijama tokena. Ova matrica se koristi za inicijalizaciju težina u prvom, embedding sloju našeg sekvencijalnog klasifikatora. Da bi naš model bio potpun, potrebno je dodati i potpuno povezani sloj sa sigmoid aktivacionom funkcijom, ali se ovom sloju prosleđuje dvodimenzioni vektor, pa je pre poslednjeg sloja potrebno dodati Flatten sloj. Ovaj model se dalje obučava i evaluira, o čemu će dalje biti reči u poglavlju sa rezultatima.

F. Klasifikacija uz pomoć Bert-a i SVM-a

U ovom eksperimentu koristili smo Bert pretrenirani model za embedding sekvence, odnosno za dobijanje matrice koju smo posle slali na SVM kao ulaz i labele koje označavaju da li je vest istinita ili lažna. Nakon učitavanja podataka, izvršili smo labeliranje svakog članka, a zatim i čišćenje članaka on nepotrebnih znakova. Skup podataka podelili smo u dva pod skupa, jedan za treniranje SVM-a, a drugi za testiranje. S obzirom da je ulaz u Bert sekvenca maksimalne dužine, bilo je potrebno prebrojati broj reči za svaki članak i implementirati pristup sliding window. Uz pomoc sliding window tehnike dobili smo veći broj članaka, međutim svi tekstovi imaju maksimalno 512 reči. Tako dobijene tekstove propustili bi kroz BertTokenizer, zatim kroz BertModel i za potrebe treniranja SVM-a koristili smo “pooler\_output” iz BertModela. Za svaki članak cuvali smo embedovanu sekvencu i labelu koja oznacava da li je članak istinit ili ne. Takav skup podataka smo dovodili na ulaz, odnosno na izlaz SVM-a i obučili model. SVM model dalje smo testirali na skupom podataka namenjenim za testiranje.

# Rezultati

Rezultati našeg sistema će biti prikazani u dve celine: rezultati podsistema za preporuku vesti i rezultati eksperimenata za detekciju malicioznih članaka.

A.   Rezultati podsistema za preporuku vesti

Evaluacija za ovaj deo sistema je izvršena nad validacionim skupom, uz pomoć microsoft recommenders metode run\_fast\_eval i uz pomoć ispisa prilikom svake iteracije, odnosno prilikom svake iteracije, računa se i evaluacija nad validacionim skupom podataka. Kao i sa svim ostalim sistemima za preporuku, javlja se problem što za svaki scenario imamo samo par tačnih odgovora i preko 99% netačnih. Iz tog razloga, za evaluaciju koristimo k negative scenarios algoritam, koji uzima jedan pozitivan ishod (korisnik kliknuo na vest) i k negativnih ishoda (korisnik nije kliknuo na vest, u našem slučaju 4) i time iteriramo više puta kroz skup pozitivnih ishoda dok se ne istroše svi negativni, čime postižemo upscaling, bez negativnog uticaja na performanse modela.

Nakon pet iteracija treninga nad demo verzijom skupa podataka, dobijena je preciznost od:  group\_auc:0.642, mean\_mrr:0.2943, ndcg@10:0.3878, ndcg@5:0.3237. Ove vrednosti možemo porediti sa vrednostima dobijenim u radu [1], uz napomenu da su u radu predstavljeni rezultati nad potpunom verzijom skupa, za koju nismo imali resurse da istreniramo naš sistem. Poređenje rezultata je prikazano na tabeli 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Naziv modela | group\_auc | mean\_mrr | ndcg@10 | ndcg@5 |
| Naš model – demo | 64.2 | 29.43 | 38.78 | 32.37 |
| LSTUR – demo | 52.01 | 22.14 | 22.92 | 29.12 |
| LSTUR | 67.73 | 32.77 | 35.59 | 41.34 |
| NPA | 66.69 | 32.24 | 34.98 | 40.68 |

Tabela 2. Poređenje našeg modela obučenog nad demo skupom, u odnosu na rezultate postojećih modela obučenih nad punom verzijom MIND skupa.

Kao što se može videti na tabeli 2, rezultati evaluacije našeg modela su skoro podjednako dobri kao i rezultati popularnih rešenja na kojima je baziran, iako je obučavan nad skupom podataka koji je više redova veličine manji od skupa koji su koristili ostali sistemi. Kada se uporedi naš model sa LSTUR modelom obučenim nad istim demo skupom, vidi se da naš model postiže group\_auc za 12% veći od osnovnog LSTUR-a.

 B.   Rezultati eksperimenata klasifikacije na istinite i lažne vesti

Evaluacija modela u svim eksperimentima je izvršena nad uzorkom od 20% početnog skupa podataka za sve modele sem BERT-a zbog manjka resursa. Kako je BERT veoma kompleksan model i zahteva dosta vremena za obučavanje i evaluaciju, uzet je uzorak od 30% početnog skupa, što je otprilike 26.000 tekstova, dok je za evaluaciju upotrebljeno 20% tog skupa. Rezultati f mera su prikazani u tabeli 3.

|  |  |
| --- | --- |
| Naziv modela | Preciznost nad testnim skupom (%) |
| LSTM | 95.4 |
| TF-IDF i Naive Bayes | 86.9 |
| GloVe i Sekvencijalna mreža | 83.0 |
| BERT i SVC | 92.7 |
| Sekvencijalna mreža | 88.4 |
| Naive Bayes | 80.0 |
| Random forest | 85.0 |
| SVM | 85.3 |
| Logistička regresija | 84.0 |

Tabela 3. Poređenje rezultata eksperimenata sa rezultatima eksperimenata iz rada [5].

Najbolji rezultati su postignuti pomoću LSTM klasifikatora, zbog karakterističnog mehanizma „zaboravljanja“ nebitnih podataka, koji je veoma bitan u ovom kontekstu, jer novinski članci sadrže veoma velik broj opisnih ili sličnih reči, koje ne nose neku težinu u okviru rečenice.

Na drugom mestu po preciznosti je BERT i support vector klasifikator, što je donekle i očekivano jer su transformeri poput BERT-a vrsta unapređenja LSTM modela. Kako je skup podataka smanjen drastično zbog kompleksnosti obučavanja ovog modela, očekujemo da bi ovaj model trebao postići i veću preciznost od LSTM-a.

Na trećem mestu se nalazi TF-IDF, koji dobija drastično bolje rezultate u odnosu na eksperiment sa Naive Bayes klasifikatorom iz rada [5] zbog mehanizma naglašavanja bitnih informacija, što dolazi do izražaja u većim tekstovima poput novinskih članaka koje koristimo ovde.

Najlošije rezultate smo postigli upotrebom glove za dobijanje embeddinga tekstova, zbog toga što glove nema mehanizam za naglašavanje bitnijih delova teksta, pa se većina procesa klasifikacije svodi na sam potpuno povezani sloj sekvencijalne mreže.

# Zaključak

Pretraga za vestima može biti bespotrebno komplikovan proces, sa obzirom na broj članaka koji se objavljuje na različitim izvorima svaki dan. Postojanje platforme za preporuku vesti bi moglo biti odlično rešenje za korisnike koji ne žele da listaju velik broj različitih sajtova i aplikacija kako bi došli do novinskih članaka koji ih zanimaju. Negativan aspekt ovakvog sistema bi bio što bi potencijalno mogao preporučivati maliciozne članke, koji bi, u najlakšem slučaju, odvlačili korisnika od korišćenja ovakve aplikacije.

Ovaj problem je podeljen u dve celine, problem preporuke vesti i problem filtriranja lažnih članaka. Problem preporuke vesti je rešen kombinovanjem NPA i LSTUR modela, baziranih na microsoft recommender sistemima i radovima [3] i [4]. Filtriranje lažnih članaka je implementirano kroz više eksperimenata, kako bi se utvrdile prednosti i mane svakog pristupa.

Najveća prednost našeg pristupa za preporuku vesti je sposobnost da kategorizuje interesuje na dugotrajne i kratkotrajne, kao i jaka povezanost sa istorijatom pretrage korisnika, što rezultuje u boljim preporukama što više korisnik interaguje sa platformom. Ovo je ujedno i mana našeg sistema, jer što manje korisnik ima interakcije sa sistemom, to će embedding članaka i korisnika biti lošiji i samim time model manje precizan.

Prostor za napredak u našem projektu je moguć kroz obučavanje modela za preporuku vesti nad punim MIND skupom, kao i izmenom Personalized Attentive Pooling mehanizma sa transformer modelima poput UniLM ili sličnih rešenja, iz razloga što se mogu postići i veće preciznosti, ali sa umanjenjem zavisnosti modela od istorije pretrage korisnika. Napredak u modelima za filtriranje je moguć upotrebom veće varijacije BERT-a, poput DeBerta XXL i sličnih modela koji imaju drastično bolji vektorizator od našeg modela.

# Literatura

1. Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. [MIND: A Large-scale Dataset for News Recommendation](https://aclanthology.org/2020.acl-main.331). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.
2. Maramreddy, Yogi Reddy & Gokul, Prathin & Asamani,. (2021). News Recommendation System. The International journal of analytical and experimental modal analysis. 13. 809.
3. Wu, Chuhan & Wu, Fangzhao & An, Mingxiao & Huang, Jianqiang & Huang, Yongfeng & Xie, Xing. (2019). NPA: Neural News Recommendation with Personalized Attention. 10.1145/3292500.3330665.
4. Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural News Recommendation with Long- and Short-term User Representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 336–345, Florence, Italy. Association for Computational Linguistics.
5. Winkelmann, Sebastian & Yousefi, Shakir & Fabricius-Bjerre, Frederik. (2020). FAKE NEWS DETECTION.

1. “MIND – Microsoft News Dataset” [Online]. Dostupno na: https://msnews.github.io/. [Objavljeno: 2020.].

1. “Fake and Real news dataset” [Online]. Dostupno na: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset. [Objavljeno: 2020.].

1. “All the news” [Online]. Dostupno na: https://www.kaggle.com/datasets/snapcrack/all-the-news. [Objavljeno: 2017.].

1. “Fake news” [Online]. Dostupno na: https://www.kaggle.com/datasets/hassanamin/textdb3. [Objavljeno: 2017.].

1. “How to generate an LDA Topic Model for Text Analysis” [Online]. Dostupno na: https://yanlinc.medium.com/how-to-build-a-lda-topic-model-using-from-text-601cdcbfd3a6. [Objavljeno: 2018.]
2. “Multi-Class Text Classification with LSTM”, [Online]. Dostupno na: https://towardsdatascience.com/multi-class-text-classification-with-lstm-1590bee1bd17. [Objavljeno: 2019.]