Klasifikacija tipa sporočil spletne diskusije o knjigah

Adam Prestor, Lojze Žust
ap2408@student.uni-lj.si, lojze.zust@student.uni-lj.si

1 Uvod

V slovenskem prostoru pismenost in želja po branju med mlajšimi generacijami počasi upadata, zato so se pojavile razne iniciative, ki želijo ta trend obrniti in ponovno oživeti zanimanje za branje in ohranjanje slovenskega jezika. Med njimi je tudi obširen projekt v katerem sodeluje več ustanov, med drugim tudi nekaj fakultet iz Univerze v Ljubljani. Cilj projekta je združiti sodobno tehnologijo in knjižno gradivo, ter tako mladim generacijam približati branje. Pomemben del projekta predstavlja orodje IMapBook. Gre za spletni prikazovalnik digitalnih knjig in ostalih gradiv, ki pa ima nekaj močnih dodatnih funkcionalnosti. Omogoča namreč izvedbo spletnih diskusij o posameznem gradivu, kjer bralci lahko svoja vprašanja in mnenja delijo z ostalimi.

V tej seminarski nalogi se osredotočamo na podatke, ki so jih v okviru projekta pridobili z raziskavo na več Ljubljanskih osnovnih šolah. Učenci so v IMapBooku prebrali pripravljeno gradivo, nato pa so se udeležili diskusije, ki je za izhodišče postavila odprto vprašanje na podlagi prebranega gradiva. Učenci so se lahko pogovarjali o temi, pri tem pa so jih usmerjali moderatorji (v tem primeru učitelji/ce). Brez posredovanja moderatorjev pogovori zelo hitro zaidejo izven teme knjige ali postanejo žaljivi. Zato se je pojavila želja, da bi sistem pomagal moderatorjem in zaznal, v kateri točki pogovora je potrebno posredovanje. V tem delu želimo nasloviti to potrebo in razviti metodo, ki bo v pomoč moderatorjem.

V tem delu naslavljamo podproblem in sicer klasifikacijo tipa sporočil – posamezno sporočilo želimo razvrstiti v eno izmed podanih kategorij (npr. pogovor o temi, klepetanje, ...). Pproblem je bolj fokusiran in jasno zastavljen, ter lažje merljiv, zato omogoča enostavnejše primerjanje razvitih metod. Dobra klasifikacija sporočil, omogoča tudi informiranje moderatorja o morebitnem potrebnem posredovanju. Če na primer nekaj časa zelo majhen del sporočil predstavlja sporočila o dejanski temi, potem se lahko izda obvestilo moderatorju, da preveri vsebino.

Problem je zelo zahteven zaradi specifik spletnih klepetalnic. Sporočila so zelo kratka in vsebujejo ogromno količino slovničnih napak in slogovno zaznamovanih besed, kar otežuje tokenizacijo in ostale dele običajne analize besedil. Poleg tega se sporočila zelo pogosto sklicujejo in nanašajo druga na drugo, uporabljajo se uporabniška imena in imena naučena tekom pogovora. Kontekst sporočila je tako zelo širok in ga je težko jasno določiti. Zbrani podatki temelijo na diskusijah o zgolj treh različnih delih, kar lahko povzroči slabo generalizacijo na ostala dela. Metoda se lahko na primer nauči, da so pogovori, kjer se omenja cefizlja vredu, vendar to glede na kontekst ni nujno res.

2 Sorodna dela

Obstaja več različnih pristopov za klasifikacijo sporočil. V grobem bi jih lahko razbili na dva dela – tiste, ki sporočila klasificirajo neodvisno drug od drugega in tiste, ki upoštevajo zaporedje sporočil pri klasifikaciji. Pri predpostavki neodvisnih sporočil, lahko posamezno sporočilo obravnavamo kot en element učne množice v klasičnih algoritmih strojnega učenja (naključni gozdovi, SVM, ipd.).

Ker je odvisnost med sporočili znotraj posameznega pogovora zelo visoka – pogosto s na primer sporočila odgovor na vprašanje v predhodnjih sporočilih – bi boljši model moral biti sposoben upoštevati tudi ostala sporočila v pogovoru. Če obravnavamo pogovor kot sekvenco sporočil, lahko uporabimo metode za označevanje sekvenc, ki poskušajo za klasifikacijo posameznega elementa upoštevati tudi sosednje elemente. Taka metoda so skriti Markovi modeli (Rabiner, 1989), ki pa jih v našem primeru ni smiselno uporabiti,

saj je množica opazovanih stanj (vsa možna sporočila) neskončna. Tudi z izločanjem značilk iz besedila težko smiselno omejimo prostor, da bi uporabili ta pristop. Uporabimo pa lahko metodo CRF (Lafferty et al., 2001), ki namesto množice opazovanih stanj uvedejo prostor značilk, na podlagi katerih se izračuna prehodna verjetnost. Vsak element sekvence moramo tako predstaviti s točko v tem prostoru značilk. Značilke lahko izluščimo ročno (prisotnost določenih besed, ločil, dolžina, itd.), ali pa z uporabo vložitev. Kot bolj napredne metode je vredno omeniti tudi globoke metode – rekurenčne nevronske mreže (LSTM (Hochreiter and Schmidhuber, 1997)) in transformerje (Vaswani et al., 2017). Trenutno dajejo najboljše rezultate na širokem spektru problemov in so sposobne modeliranja kompleksnih in daljših odvisnosti znotraj sekvenc.

Nekaj del (Weisz; Dong et al., 2006) se ukvarja specifično s problemom klasifikacije teme pogovora v kontekstu spletnih klepetalnic in uporabljajo podobne metodologije.

3 Metodologija

Procesiranje in klasifikacija poteka v več korakih. V prvem koraku naredimo predprocesiranje sporočil. V tem koraku se izvede robustna tokenizacija sporočil, in priprava bag-of-words značilk. Pripravimo tudi nekaj dodatnih značilk, ki opisujejo vsebino in kontekst sporočila. Za napovedovanje tipa sporočila na podlagi pridobljenih značilk smo uporabili več različnih metod - naključne gozdove, model SVM, in CRF-je za označevanje sekvenc.

3.1 Predprocesiranje sporočil

Zaradi kratke dolžine in nezanesljivosti črkovanja v sporočilih, je robustno predprocesiranje zelo pomemben vidik obdelave vhodnih podatkov. Ker je število besed zelo majhno smo si za cilj zadali, da združimo kar se da veliko število besed z istim pomenom. Prvi korak predprocesiranja je tokenizacija. V ta namen smo uporabili modul *TweetTokenizer* iz knjižnjice *nltk*. Ta je prilagojen za tokenizacijo čivkov, ki so vsaj malo sorodni sporočilom. Ravno tako so krajše oblike in niso slogovno konsistentni. Potrebna je bila še dodatna ročna tokenizacija v primerih, kjer za stavčnim ločilom ni bilo presledka. Tokene, ki so bili sestavljeni iz besedila in številk smo posebej razbili na besede in številke.

V drugem delu predprocesiranja odstranimo stop besede. Uporabili smo slovar slovenskih stop besed , ki smo ga dobili na github repozitoriju. Ker so sporočila že tako precej kratka smo se odločili slovar stop besed nekoliko zmanjšati in ohraniti nekaj besed, ki so se nam zdele informativne za napovedovanje tipa sporočil. Tako smo iz slovarja stop besed odstranili večino vprašalnic (kaj, kako, zakaj, ...), saj te dajejo pomembno informacijo o tipu sporočila. V tretjem delu izvedemo lematizacijo. S tem želimo zmanjšati število različnih pojavitev iste besede. Uporabili smo lematizator *Lemmagen*. V naslednjem koraku iz besed odstranimo vse šumnike. Pretvorimo jih v ustrezne nešumne različice (npr. š v s). Velik del uporabnikov je namreč besede pisal brez šumnikov. S tem postopkom združimo iste besede pisane brez ali s šumniki.

V enem izmed korakov predprocesiranja smo želeli odpraviti tudi tipkarske napake in pogovorno rabo besed. V besedilih se namreč pojavi veliko besed s tipkarskimi napakami. Za detekcijo napak smo skušali uporabiti slovar slovenskih besed in Levenstheinovo razdalje. Ideja je bila, da poiščemo najbližjo besedo v slovarju, in če je razdalja dovolj majhna izvedemo popravek. Žal je metoda delovala prepočasi, da bi jo uporabili, zato smo za razdaljo uporabili *difflib*. Ta metoda je delovala precej hitreje, žal pa zaradi načina ocenjevanja razdalje rezultati niso bili najboljši. Metoda je pravilno popravila nekaj besed, napravila pa je tudi veliko napačnih menjav. Iz teh razlogov smo se na koncu odločili, da metode ne vključimo v postopek predprocesiranja.

Zadnji korak predprocesiranja je združevanje določenih tokenov v skupine. Tako smo združili vse številske tokene v skupni token <number>, vse tokene ki vsebujejo neznane znake v token <other> in nesmiselne tokene v token <gibberish>. Med pregledom sporočil smo ugotovili, da velik del sporočil predstavljajo nesmiselna spam sporočila (naključni pritiski na tipke). Za detekcijo nesmislov smo implementirali preprosto Markovo verigo, ki smo jo naučili na slovarju slovenskih besed. Markov model je modeliral verjetnost prehodov med pari znakov. Ko je bil model naučen smo z uporabo ročno izbranih predstavnikov dobrih in slabih sporočil določili mejo verjetnosti, ki ločuje dobra sporočila od nesmislov. Z uporabo tega modela smo avtomatsko detektirali veliko večino nesmiselnih sporočil in jim dodelili token <gibberish>.

3.2 Priprava značilnic

Na podlagi tokenov pridobljenih na zgoraj opisan način smo pripravili značilnice sporočil z uporabo modela bag-of-words. Najprej smo zgradili slovar tokenov, ki bodo uporabljene v značilkah. Uporabili smo 512 najpogostejših tokenov v slovarju. Osnovni BoW model uporablja absolutne frekvence (število pojavitev) besed v sporočilu, pri izboljšanem modelu pa smo uporabili utežitev tf-idf.

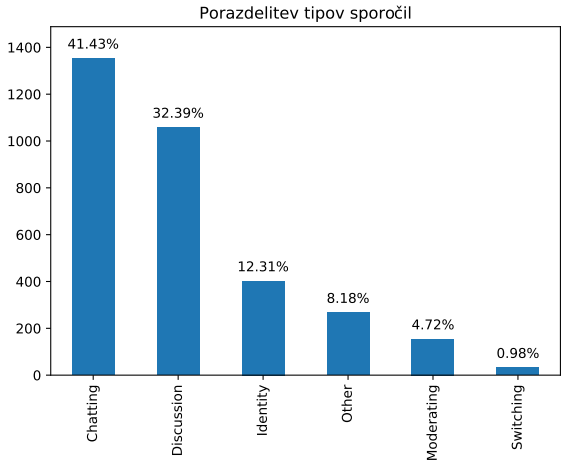
BoW smo zgradili posebej tudi za besedila iz knjig in tem pogovora. Te smo uporabili, ko smo poskušali ugotoviti podobnost sporočila z knjigo in/ali s temo pogovora. Več o tem bomo spregovorili v naslednjem poglavju, kjer bomo predstavili ročne značilke.

Poleg BoW modela smo iz sporočil, brez predprocesiranja, sestavili značilke iz bigramov. Kot poseben token za presledke smo uporabili znak "_". Iz dobljenih bigramov za vsa sporočila smo zgradili slovar tokenov, ki so bile uporabljene v značilkah. Uporabili smo 1024 najpogostejših bigramov. Bigram model uporablja število pojavitev zaporednih črk v sporočilu. Daljših ngramov nismo vzeli, ker bi bilo število značilnic preveliko.

3.2.1 Ročne značilnice

Poleg zgoraj naštetih značilk, pa smo nekatere značilke določili ročno.

- **Dolžina sporočila** - število znakov v sporočilu.
- **Število besed** - število besed v sporočilu. Stop besede ipd. smo obdržali.
- **Nedavna aktivnost** - v določenem časovnem intervalu (5 minut) pred trenutnim sporočilom preštejemo število vseh sporočil v pogovoru, število sporočil trenutnega uporabnika in število različnih uporabnikov.
- **Sentiment** - na podlagi slovarja besed s pozitivnim in negativnim sentimentom besed ocenimo sentimentno vrednost sporočila. Pozitivne besede prištejejo 1, negativne pa odštejejo 1 končni oceni. Končna ocena se normira z dolžino sporočila, tako da je ocena sentimenta med -1 in 1. Slovar sentimenta smo pridobili na strani Clarin.si
- **Sorodnost knjigi** - preštejemo število besed v sporočilu, ki se pojavijo v knjigi, na katero se pogovor nanaša.



Slika 1: Porazdelitev števila sporočil v podatkovni zbirki glede na tip sporočila. Zbirka je v glavnem sestavljena iz sporočil tipa Chatting in Discussion.

- **Podobnost s tematiko** - primerjamo BoW vektorja sporočila in tematike in izračunamo medsebojno kosinusno razdaljo.
- **Podobnost s prejšnjim sporočilom** - koliko besed iz prejšnjega sporočila se pojavi v trenutnem.

3.3 Napovedni modeli

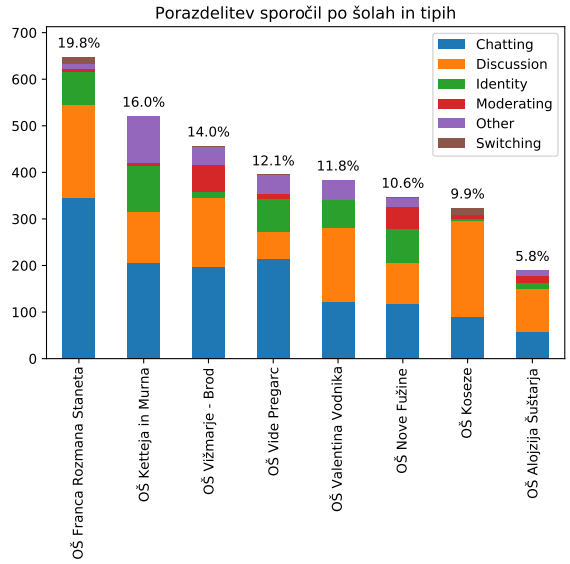
Za napovedovanje smo uporabili več različnih modelov. Kategorijo posameznih sporočil smo napovedovali z metodami strojnega učenja, ki ne upoštevajo zaporedja posameznih sporočil, kot so naivni Bayes (NB), naključni gozd (RF), odločitveno drevo in metoda podpornih vektorjev (SVM). Uporabili pa smo tudi CRF označevalni model, ki posameznim sporočilom pripiše oznako glede na značilnice, pri tem pa upošteva tudi ostala sporočila v zaporedju.

4 Rezultati

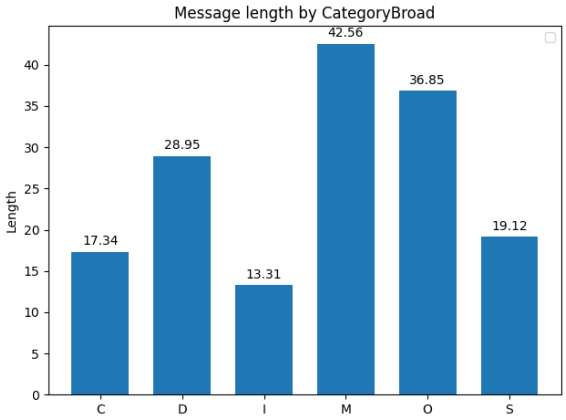
4.1 Analiza podatkovne zbirke

Pred začetkom implementacije smo izvedli preliminarno analizo podatkovne zbirke, da bi pridobili čim več informacij, ki bi nam pomagale pri izbiri ustrezne metodologije.

Najprej smo preverili porazdelitev števila sporočil med posameznimi tipi sporočil (glej Sliko 1). Največji del sporočil predstavljata kategoriji Chatting, ki predstavlja neformalen pogovor in Discussion, ki predstavlja pogovor o knjigi. Predvidevamo, da bo ti dve kategoriji tudi najtežje ločiti,



Slika 2: Porazdelitev števila sporočil po šolah. Prikazana je tudi porazdelitev tipov sporočil znotraj posamezne šole, ter delež sporočil, ki jih posamezna šola doprinese v celoto.



Slika 3: Povprečna dolžina sporočila (v znakih) glede na tip sporočila. Vidne so velike razlike v povprečni dolžini med posameznimi tipi sporočil. (C: Chatting, S: Switching, D: Discusion, M: Moderating, O: Other, I: Identity)

saj so sporočila podobne strukture. Izmed ostalih tipov je največ Identity, ki vsebuje pogovore o identiteti posameznikov.

Na Sliki 2 je prikaz števila sporočil posameznih tipov po različnih šolah v zbirki. Vidimo da se število sporočil precej razlikuje med šolami. Prav tako je v nekaterih šolah več neformalnega pogovora, drugje pa so se učenci bolje držali teme.

Na Sliki 3 smo analizirali še povprečno dolžino sporočila po posameznih kategorijah. Vidimo da so sporočila moderatorja v povprečju najdaljša. Prav tako so sporočila diskusije povprečno rahlo daljša kot klepetanje. Zaradi velikih razlik med povprečnimi dolžinami sporočil smo se odločili, da dolžino sporočila uporabimo kot eno izmed značilk.

4.2 Evalvacija

4.2.1 Ločevanje na učno in testno množico

Vhodne podatke, sporočila, smo najprej razdelili glede na šolo, temo pogovora in pogovorno skupino. Tako smo dobili posamezne pogovore. Učno in testno množico pa smo naredili tako, da smo izbrali eno izmed šol. Vsi pogovori, ki so bili s te šole, smo upoštevali kot testno množico, vsi ostali pogovori pa so bili del učne množice.

Zakaj takšna delitev? Ker s tem dobimo najbolj neodvisno delitev med pogovori, saj bi lahko pogovori znotraj šole vplivali eden na drugega in tako ne bi dobili tako zanesljivih rezultatov. Hkrati taka delitev najbolje predstavlja realno aplikacijo algoritma, ki mora biti dovolj robusten, da deluje neodvisno od množice sodelujočih oseb.

Opazili smo tudi, da je imela podana podatkovna baza nekaj tiskarskih škratov pri naslovu šol, ki smo jih tekom implementacije odpravili. Tako smo dosegli, da so bile množice resnično ločene po posameznih šolah.

4.2.2 Križna validacija

Vsi rezultati, ki jih bomo predstavili, so bili dobljeni s križno validacijo. Križno validacijo smo dobili tako, da smo kot testno množico izbrali vse pogovore z ene šole, množica za treniranje pa je vse ostalo. To smo ponovili za vse šole, torej smo imeli 8 foldov. Rezultate validacij smo nato sešteli skupaj in dobili končni rezultat, ki je uravnotežena vsota vseh posameznih foldov.

Omenimo še, da so foldi bili različnih dolžin. Dolžina posameznega folda je predstavljena na Sliki 2, kjer vidimo, da niha med 700 in 200 sporočili, kar je kar precejšna razlika, zato smo morali uporabiti uteženo povprečje rezultatov.

4.2.3 Izbira značilk

Za klasifikacijo smo si pripravili kar precej različnih značilk. Zanimalo nas je, katere značilke so tiste, ki so sploh uporabne za učenje napovednih modelov?

Kar hitro se je videlo, da so značilke, ki opisujejo sorodnost s knjigo ali podobnost z tematiko pogovora, precej neuporabne, saj se obe pre-

več usmerita na največji razred, klepetanje. Tudi v kombinaciji z ostalimi značilkami ponavadi poslabšata rezultat.

Izpustili smo tudi značilko podobnosti s prejšnim sporočilom. Zanimivo je, da je ta značilka največ pripomorla k tem, da smo napovedovali razred drugo. Razlog za izpustitev je, da slednja značilka ni najbolj primerna za modele CRF, ki povezanost sporočil nekako že upoštevajo pri izgradnji modela.

Iz vseh ostalih značilk pa smo zgradili 3 različne skupine, ki smo jih uporabili za učenje in napovedovanje.

- dolžina sporočila, število besed, sentiment in nedavna aktivnost (ročne značilke)
- značilke BoW tf-idf in bigrami
- vse značilke skupaj

Zakaj takšna delitev? Prva skupina nam pove, kako natančni smo lahko, če uporabimo nekaj ročno generiranih značilk, ki niso direktno povezane z obdelavo naravnega jezika. Razlog za drugo skupino je ravno nasproten. Tretja skupina pa je tam zato, da ugotovimo, ali kombinacija obeh skupin lahko privede do najboljših rezultatov. Glede na vmesne rezultate, kjer smo združili značilke BoW tf-idf in število besed, smo namreč dosegli višji rezultat kot le z BoW tf-idf značilkam.

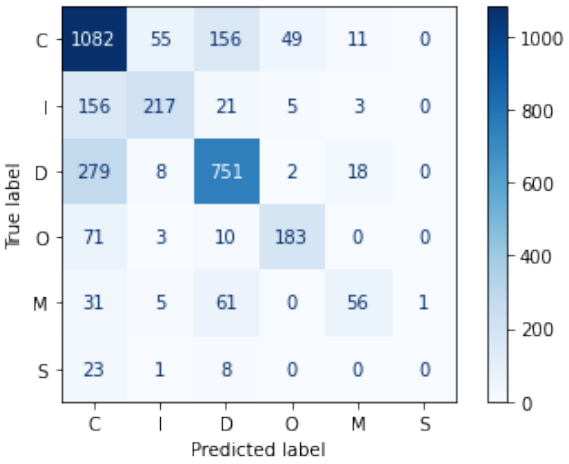
4.2.4 Izbira napovednih modelov

Izbor napovednih modelov je bil predvsem s stališča modelov, ki ne upoštevajo vrstnega reda sporočil. Izbirali smo med NB, odločitvenimi drevesi, RF in SVM. Izbrali smo modela, ki sta se najbolj izkazala tekom priprav značilnic. Odločili smo se za RF in SVM, ki sta bila mnogo boljša od vseh ostalih.

Za pripravo rezultatov smo tako uporabili modele SVM, RF in CRF.

4.2.5 Analiza rezultatov

Pognali smo križno validacijo na zgoraj opisanih modelih zbirkah značilk. V Tabeli 4.2.5 so prikazane F-ocene posameznih eksperimentov. Najboljši model smo podrobneje analizirali in izračunali metrike za posamezne razrede (glej Sliko 4 in Tabelo 4.2.5). Model precej dobro napove kategorije Chatting, Identity, Discussion in Other. Moderating in Switching napove precej slabše. To je deloma razumljivo, saj ti podatki predstavljajo zelo majhen delež učne množice (glej Sliko 1). Matrika razvrščanja razkrije, da se največ napak pojavi pri razločevanju med Chatting in Discussion kategorijama. V nadaljnem delu bi se bilo vredno fokusirati na izboljšavo tega dela.



Slika 4: Matrika razvrščanja na testnih podatkih za najboljši napovedni model CRF. Model precej natančno napove najpogostejše 4 kategorije. (C: Chatting, S: Switching, D: Discusion, M: Moderating, O: Other, I: Identity)

	F-ocena		
	SVM	**RF**	**CRF**
ročne značilke	0.4492	0.4950	0.4936
BoW + bigrami	0.6221	0.6462	**0.6932**
vse značilke	0.6309	0.6534	0.6927

Tabela 1: F-ocene različnih modelov in značilk z uporabo križne validacije po šolah. Najbolje se je obnesel model CRF z uporabo BoW in bigramov.

5 Zaključek

Tekom semestra smo izdelali program, ki napoveduje tip sporočil, ki jih dobimo iz sistema iMap. V implementaciji smo uporabili metode predprocesiranje, kjer smo popravljali slovnične napake, loče-

	Pr	**Re**	**F1**
C	0.6590	0.7997	0.7225
I	0.7509	0.5398	0.6281
D	0.7458	0.7098	0.7274
O	0.7657	0.6854	0.7233
M	0.6364	0.3636	0.4628
S	0.0000	0.0000	0.0000

Tabela 2: Natančnost (Pr), priklic (Re) in F-ocena (F1) po posameznih razredih za najboljši model.

vali sporočila na žetone, odstranjevanje stop besed in lematizacijo. Lotili smo se izluščevanja značilk, s pomočjo metod procesiranja naravnega jezika, kot tudi nekaterih ročno izluščenih, kot je na primer podobnost s knjigo, podobnost s prejšnim sporočilom in pa število besed. Podatkovno zbirko smo razdelili na testno in učno množico in uporabili križno validacijo za evalvacijo rezultatov. Za napovedovanje smo testirali in uporabili napovedne modele, kot so RF, SVM in CRF. Uspešnost medolov smo ocenili z F-oceno, zgradili pa smo tudi konfuzijsko matriko ter izračunali natančnost, priklic in F-oceno za vsak razred in celoto.

Ugotovili smo, da so modeli, ki upoštevajo zaporedje sporočil, bolj natančni. To se nam zdi smiselno, saj se zaporedna sporočila v večini nanašajo ena na drugo.

Poleg tega so se značilke, kot so BoW in ngrami, izkazale kot bistveno boljše za napovedovanje razreda. Kljub vsemu, pa so se nekatere ročne značilke izkazale za precej vredu, kot sta dolžina in število besed.

Zanimiva ugotovitev je tudi, da podobnost knjigi in temi ni bil bolj učinkovit za razlikovanje med klepetanjem in diskusijo.

6 Nadaljne delo

Velik potencial za izboljšavo rezultatov ima uravnoteževanje učne množice. Razredi sporočil so precej neuravnoteženi med seboj, kar 74% vseh sporočil spada med klepetanje ali diskusijo, medtem ko je spremembe teme manj kot 1%. Težavo bi lahko rešili s pretiranim vzorčenjem ostalih razredov. Tako bi se izognili situaciji, ki je nastala, da se značilke preveč prilagodijo večinskemu razredu.

Poleg tega bi lahko za izluščenje značilk vzeli pristope globokega učenja, npr. z uporabo rekurenčnih nevronskih mrež in transformatorjev, ki trenutno dajejo najboljše rezultate.

Možno bi bilo tudi dodajanje novih ročnih značilk, ki bi bolje opisovale posamezne razrede. Nekatere, ki smo jih uporabili, so se izkazale za precej uporabne, kot so dolžina sporočila in število besed. Žal smo z drugimi malo brcnili v temo, vendar smo optimistični, da obstajajo še kakšne, ki bi pozitivno prispevale k točnosti napovedovanja.

Viri

Haichao Dong, Siu Cheung Hui, and Yulan He. 2006. Structural analysis of chat messages for topic detection. *Online Information Review*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Justin Weisz. Segmentation and classification of online chats.