Klasifikacija sporočil spletnega pogovora 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). Opazimo lahko, da 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. Osredotočamo se na podproblem in sicer klasifikacijo posameznih sporočil v različne kategorije. Podproblem je bolj fokusiran in jasno zastavljen, ter lažje merljiv in omogoča enostavnejše primerjanje razvitih metod. Poleg tega taka klasifikacija omogoča osnovno 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 ni enostaven zaradi specifik spletnih klepetalnic. Sporočila so zelo kratka in vsebujejo ogromno količino slovničnih napak in slangovskih besed, kar otežuje tokenizacijo in ostale dele analize besedil. Poleg tega se sporočila zelo pogosto sklicujejo 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 besedil. Osnoven model za klasifikacijo je naivni Bayes, ki predpostavlja neodvisnost in nepomembnost vrstnega reda besed in deluje na podlagi pogojnih verjetnosti, da se beseda w pojavi v besedilu tipa c. Za tak pristop ne pričakujemo velikega uspeha, saj je povprečno število besed v besedilih zelo majhno, ter z njim ne zajamemo konteksta pogovora. Bi pa tak model mogoče znal detektirati sporočila, ki niso relevantna za temo (npr. iskanje identitete, kletvice).

Boljši model bi moral biti sposoben upoštevati 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. Uporabimo pa lahko pogojne Markove modele (MEMM) (McCallum et al., 2000) in metode 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.

Za značilke lahko vložitve pridobimo na več načinov. Lahko uporabimo redek model bag-of-words, kar pa za metode označevanja sekvenc ni preveč uporabno, saj je s tem načinom število značilk preveliko. Bolj uporabne so goste vložitve, ki bistveno zmanjšajo število dimenzij. Tu pridejo v poštev metode kot so PCA (Jolliffe, 1986), word2vec (Mikolov et al., 2013) in globoke vložitve. V primeru uporabe globokih vložitev, je zanimiva tudi možnost izgradnje end-to-end arhitekture, ki združuje globoke vložitve in klasifikacijo sekvenc v enovit model, kjer lahko obe komponenti optimiziramo hkrati. Tak model zna biti vprašljiv zaradi relativno majhne količine podatkov.

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

3 Metodologija

Procesiranje poteka v več korakih. V prvem koraku naredimo predprocesiranje sporočil. V tem koraku se izvede robustna tokenizacija sporočil in priprava BoW značilk. Iz besedila izločimo tudi nekaj ročnih značilk. Za napovedovanje tipa sporočila smo uporabili več različnih metod - navadne pristope strojenga učenja, Markove modele 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 tweetov, ki so vsaj malo sorodni sporočilom, kot jih imamo mi. 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.

Drugi del predprocesiranja je odstranjevanje stop besed. Uporabili smo slovar slovenskih stop besed , ki smo ga dobili na github repozitoriju. Ker je število besed že tako precej majhno smo se odločili obdržati nekaj besed, ki so se nam zdele informativne za napovedovanje tipa sporočil. Tako smo iz slovarja stop besed odstranili večino vprašalnic, saj so te pomembne pri ločevanju vprašanj od odgovorov. V tretjem delu smo se lotili lematizacije. S tem smo želeli zmanjšati število različnih pojavitev iste besede. Uporabili smo lematizator *Lemmagen*. V nadaljnem koraku se iz besed odstranijo vsi šumniki, tako da jih pretvorimo v nešumne različice (npr. š v s). Velik del sporočil namreč predstavljajo sporočila brez šumnikov, zato bi v teh primerih obstajlo več verzij istih besed.

V enem izmed korakov predprocesiranja smo želeli odpraviti tudi tipkarske napake in pogovorno rabo besed. To smo poskušali z uporabo slovarja slovenskih besed in uporabo Levenstheinove 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 končno različico.

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 verjetno-

sti, 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 izgradili slovar tokenov, ki bodo uporabljene v značilkah. Tu smo uporabili 511 najpogostejših tokenov, preostale tokene pa smo združili v poseben token <OOD>, ki označuje besede izven slovarja. Značilnice imajo tako 512 dimenzij. Osnovni BoW model uporablja absolutne frekvence (število pojavitev) besed v sporočilu, pri izboljšanem modelu pa smo uporabili utežitev tf-idf. Poleg BoW značilnic smo iz sporočil izločili še ročne značilnice. Trenutno smo tu uporabili samo število besed v sporočilu.

3.3 Napovedni modeli

Za napovedovanje smo uporabili več različnih modelov. V prvem sklopu smo problem obravnavali kot klasifikacijo posameznih sporočil (brez upoštevanja zaporedja znotraj pogovora). Kategorijo posameznih sporočil smo napovedovali z metodami strojnega učenja, kot so naivni Bayes, naključni gozd, odločitveno drevo in metoda podpornih vektorjev. V drugem sklopu smo uporabili CRF označevalni model, ki posameznim sporočilom pripiše oznako glede na značilnice, pri tem pa upošteva tudi ostala sporočila v zaporedju.

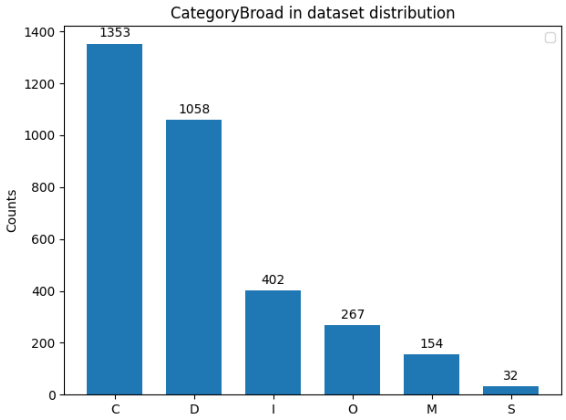
4 Rezultati

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. 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



Slika 1: Porazdelitev števila sporočil v podatkovni zbirki glede na tip sporočila. (C: Chatting, S: Switching, D: Discusion, M: Moderating, O: Other, I: Identity)

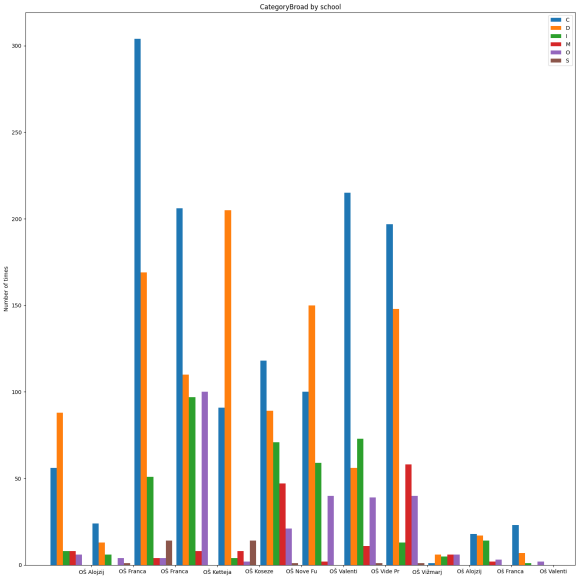
tako je v nekaterih šolah več neformalnega pogovora, drugje pa so se učenci bolje držali teme.

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. Nekatere razlike so precejšnje, tako da bi dolžino sporočila mogoče lahko uporabili kot uporabno značilko pri napovedih.

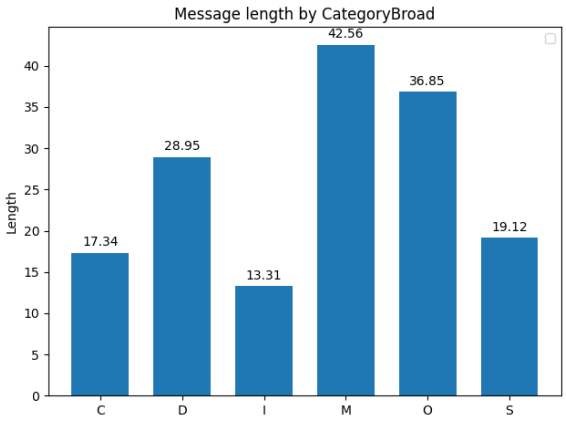
4.2 Evalvacija

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 sposobna generalizacije na različne šole. Pri nekaterih eksperimentih smo uporabili križno validacijo, tako da smo vsakič eno šolo pustili v testni množici, ostale pa so bile v učni množici. Pri tem je potrebno poudariti, da v zadnjem trenutku opazili, da posamezne šole v datasetu niso označene z enoličnimi oznakami, tako da naša delitev ni povsem ločena na posamezne šole. To bomo odpravili v naslednjem koraku.



Slika 2: Porazdelitev tipa sporočil po različnih šolah. (C: Chatting, S: Switching, D: Discusion, M: Moderating, O: Other, I: Identity)



Slika 3: Povprečna dolžina sporočila (v znakih) glede na tip sporočila. (C: Chatting, S: Switching, D: Discusion, M: Moderating, O: Other, I: Identity)

4.2.2 Strojno učenje

Vsi rezultati so bili narejeni za značilke BoW. Rezultati so bili pridobljeni tudi glede na dolžino sporočil, kjer so se gibali okoli 55% z manjšimi odstopanji glede na izbran klasifikator.

Naivni bayes, katerega rezultati so vidni na Sliki 4, nam je podal najslabšo napoved izmed vseh preizkušenih metod. Njena natančnost je bila $39,68\%$. Razlog za to je predvsem slaba napoved razprave (C), ki zavzema največji del sporočil.

Metoda najbljižjih sosedov je pokazala že bistveno izboljšavo, dosegla je namreč natančnost do $56,34\%$. Metodo smo testirali z različnim številom sosedov - 3,5,7,9. Najbolje se je odrezala pri 5.

Z metodo naključnih gozdov smo dosegli natančnost $60,10\%$. Ta metoda je predvsem pravilno napovedovala večinski razred, C, 80% in drugi največji razred, D, 54%.

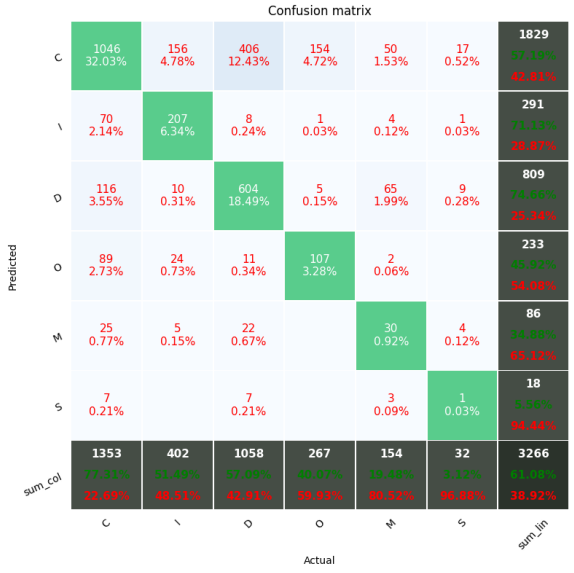
Najboljše pa se je odrezala metoda podpornih vektorjev, rezultati so vidni na Sliki 5, ki je dosegla klasifikacijsko točnost $61,08\%$. Prednost te metode pred naključnimi gozdovi je tudi v tem, da poleg pravilne napovedi tipov C in D tudi za tipa I in O ohrani solidno točnost.

Iz rezultatov je razvidno, da napovedovanje tipov sporočil M in S zelo nezanesljiva. Razlog za to je nizko število pojavitev sporočil teh tipov. Poleg tega pa je natančnost klasifikatorja predvsem odvisna od glavnih dveh tipov sporočil, C in D, ki, kot se vidi na Sliki 1, predstavljata veliko večino sporočil v zbirki.

4.2.3 CRF modeli

Pri evalvaciji CRF modelov smo testirali modele naučene na različnih množicah značilnic. To vključuje uporabo števila besed, značilnic BoW, značilnic BoW s tf-idf in kombinacijo BoW (tf-idf) in števila besed. Izvedli smo križno validacijo in za metriko vzeli povprečno natančnost križne validacije. Rezultati so prikazani v Tabeli 4.2.3. Model bag-of-words deluje solidno, ampak zaradi kratkih sporočil in velikih razlik med šolami ni vedno sposoben dobre generalizacije. Kombinacija z ročno izločenimi značilkami daje precej boljše rezultate. V prihodnje bi zato bilo smiselno poskusiti najti še več močnih ročno določenih značilk. Tu bi lahko uporabili na primer število predhodnih sporočil istega uporabnika, kakšne značilke iz teme in vsebine knjige in podobne hevristike.

Slika 4: Rezultati napovedovanja tipa sporočila z naivnim Bayesovim klasifikatorjem. Stolpci predstavljajo pravi tip vrstice pa napovedan tip sporočil. Natančnost je zapisana v spodnjem desnem kvadratu z zeleno barvo.



Slika 5: Rezultati napovedovanja tipa sporočila z metodo podpornih vektorjev. Stolpci predstavljajo pravi tip vrstice pa napovedan tip sporočil. Natančnost je zapisana v spodnjem desnem kvadratu z zeleno barvo.

model	**natančnost**
majority	0.3952
num words	0.4672
BoW	0.5951
BoW (tf-idf)	0.6056
BoW (tf-idf) + num words	**0.6381**

Tabela 1: Rezultati CRF modelov z uporabo križne validacije. Modeli uprabljajo različne značilnice. Vidimo da model bag-of-words deluje bolje od uporabe zgolj števila besed. Najboljše rezultate pa daje kombinacija obojega. Za primerjavo je prikazana tudi točnost, ki jo pridobimo z uporabo večinskega klasifikatorja.

5 Zaključek

Glede na dobljene rezultate vidimo, da ne glede na izbrano metodo dobimo rezultate, ki presegajo večinski klasifikator. Kljub temu, da so metode strojnega učenja dokaj natančne, so CRF modeli presegli njihovo natančnost. Iz tega lahko sklepamo, da zaporedje sporočil pripomore k natančnost napovedovanja. Prav tako imamo neuravnovešene razrede, saj razreda C in D predstavljata večino podatkovne zbirke.

Poleg tega smo opravili še test napovedovanja tipa sporočila z Markovimi modeli. Model smo zgradili glede na prehajanje tipov sporočil med zaporednimi sporočili. Za vsak tip smo izbrali najverjetnejši prehod za pravilnega. Metodo testiramo tako, da na podlagi prejšnega sporočila poskušamo napovedati novo. Tako smo dosegli natančnost $66,67\%$. Vendar pa je ta model nerealen, saj moramo zanj vnaprej vedeti tipe sporočil.

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.

Ian T Jolliffe. 1986. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer.

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

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.