Zaznavanje subreddita v komentarjih

Nejc Kozlevčar

Fakulteta za računalništvo in informatiko, 1000 Ljubljana, SI
nk3007@student.uni-lj.si

Povzetek Reddit je postal zelo priljubljeno družabno omrežje, kjer obstaja že ogromno tako imenovanih subredditov. V našem delu predlagamo pristop strojnega učenja, kjer na podlagi komentarjev določimo v kateri subreddit spada. Komentarje imamo iz dveh različnih subredditov, iz njih na podlagi leksikografskih, sintaktičnih, sentimentnih in semantičnih lastni zgradimo model učenja.

1 Uvod

Cilj druge naloge pri predmetu Obdelava naravnega jezika (ONJ) je, da na komentarjih iz portala Reddit, zgradimo model, ki na podlagi komentarja kar najbolje napove pripadajoč subreddit. Za reševanje problema uporabljamo pristope obdelave naravnega jezika, ki smo se jih učili pri predmetu.

Reddit je socialno omrežje, ki je v zadnjih letih ogromno pridobil na popularnosti. Reddit je skupek ogromno skupnosti imenovanih subbredit. Trenutno je aktivnih skupnosti že več kot 138 tisoč, aktivni uporabnikov pa preko 330 milijonov. Stran ja po podatkih spletne strani Alexa [1] 5 najbolj obiskana stran v Ameriki. To so zgolj nekateri podatki, ki kažejo na to, da je to socialno omrežje med uporabniki res priljubljeno [2].

Na vsaki skupnosti (ang. *subreddit*) je dnevno objavljeno več novic. Pod vsako novico pa se nabere ogromno komentarjev. Zanimalo nas je ali lahko komentarju pripišemo pripadajočo skupnost. Za učno množico smo vzeli zgolj 2 skupnosti in to sta "nba"in "politics".

2 Podatki

Kot rečeno smo za testne podatke uporabili komentarje iz portala Reddit, ki so bili napisani dne 1.1.2018. Podatke smo dobili na povezavi, kjer ponujajo tudi komentarje na mesečni ravni. Toda za naš primer je en dan zadostoval.

V dobljenih podatkih so zbrani podatki iz celotnega portala, ker je za našo nalogo bistveno preveč. Vseh komentarjev dne 1.1.2018 je 2.360.226, skupna velikost datoteke pa je znašala 1.6 GB. Odločili smo se, da se omejimo zgolj na 2 skupnosti, da je klasifikacija razreda enostavnejša. Prešteli smo komentarje vsake skupnosti in se odločili, da za analizo izberemo skupnosti "nba"in "politics". V obeh kategorijah je bilo približno 22 tisoč komentarjev, kar skupaj znese 44 tisoč testnih podatkov.

To je še vedno prevelika količina podatkov, ki bi jo naš računalnik lahko prebavil. Zato smo za naš model vzeli iz vsake skupnosti zgolj 2000 komentarjev, kar skupaj znese 4000 komentarjev. Iz komentarjev smo morali izločiti vse spletne strani, ker ne sodijo v naša analizo besedila.

Iz začetnih podatkov smo vzel zgolj ime subreddita, ki smo ga kasneje pretvorili v 0 za "nba"in 1 za "politics". Ta podatek smo uporabili za označitev komentarja (ang. *label*). Drugi podatek, ki smo ga uporabili pa je bil seveda celoten komentar, ki smo ga kasneje obdelovali. Zgoraj omenjena podatkoma smo dodali še id komentarja in vse skupaj ločili s tabulatorjem.

3 Metode

Podatke opisane v zgornjem razdelku smo nato uvozili v naš program. Podatke je bilo pred začetkom gradnje modela obdelati. Kot glavni pristop preprocesiranja smo uporabili tokenizacijo. Tokenizacija je proces ločevanja besed iz povedi. Kot rezultat dobimo ločene besede iz povedi. Besede ne ločimo zgolj po presledkih, toda tudi po drugiš ločilih kot so -"ali "/"... V naši nalogi smo za tokenizacijo uporabili metodo "word_tokenize"iz knjižnice nltk.

```
for comment_id, text in self.text.items():
    self.tokens[comment_id] = word_tokenize(text)
```

Nato smo uporabili metodo TF-IDF vektorizacijo, kjer smo dobili frekvenco uporabljenih besed v besedilu, kar smo kasneje uporabili za klasifikacijo komentarjev. TF-IDF vektor nam pokaže katere besede so najbolj pogoste uporabljene v določenem subredditu [3]. Ker je v tem vektorju veliko vrednosti enakih 0 smo za implementacijo uporabili sparse matriko, da prihranimo s pomnilnikom. Za implentacijo IF-IDF smo uporabili razderd TfidfVectorizer iz sklearn knjižnice.

```
vect = TfidfVectorizer(strip_accents="unicode",
    analyzer="word", stop_words="english")
X = csr_matrix(vect.fit_transform(self.text.values()))
```

Za izboljšanje napovednega modela smo zgornjemu modelu dodali še pogostost parov besed in trojčkov v besedilu. Torej, da pogledamo tudi kako pogoste so določne besedne zveze in tri povezane besede. Dobljeni matriki smo nato dodali prejšnji in rezultat klasificirali s pomočjo razreda LinearSVC iz knjižnice SkLearn.

4 Rezultati

Za validacijo naučenega modela smo uporabili 3-kratno navkrižno validacijo (*3-fold cross validation*. Za primerjavo smo uporabili tudi naključni klasifikator, kjer smo ugibali ali je bil komentar napisan na nba ali politics subreditu.

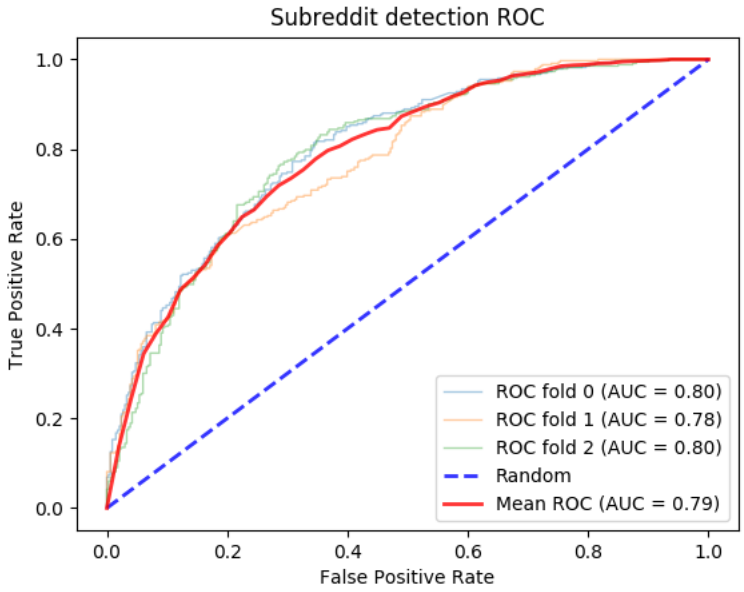
Za izračun smo uporabili 4 merske enote:

- Accuracy = št. pravilno napovedanih napovedi / št. vseh napovedi
- Precision = št. pravilno napovedanih pozitivnih napovedi / št. napovedanih pozitivnih napovedi
- Recall = št. pravilno napovedanih pozitivnih napovedi / št. vseh pozitivnih napovedi
- F-score = 2 (Precision * Recall) / (Precision + Recall)

	Random classifier	My prediction
Accuracy	0.501	0.661
Precision	0.501	0.819
Recall	0.495	0.412
F-score	0.498	0.548

Tabela 1. Rezultati

Vseh komentarjev je nemogoče pravilno umestiti v pravilni subreddit, saj so komentarji lahko v vseh subreddit enaki. Tukaj predvsem mislim na splošne komentarje, ki sami po sebi niso vezani na temo. Taki si naprimer: "That's a joke?", "Oh come on.",... Rezultati pa so vseeno nekoliko višji kot pri ugibanju, tako da lahko sklepamo da je model dobro zasnovam. Bi pa lahko z dodatno analizo rezultat še izboljšali. Za konec smo izrisali še ROC graf.



5 Zaključek

V naši nalogi smo uporabili nekaj osnovnih principov analize besedila. Po pregledu podatkov lahko ugotovimo da je model pravilno zastavljen, bi pa lahko s pomočjo dodatnih pristopov še izboljšali naš rezultat. Ugotovili smo da je tovrstna analiza zelo počasna za procesiranje, zato je velika količina podatkov lahko problematična. Ugotovili smo tudi da je analiza komentarjev težavna, ker se v veliki večini uporablja sleng, ki pa v knjižnicah ni podprt.

Literatura

1. Top Sites in United States, `https://www.alexa.com/topsites/countries/US`. Last accessed 12.1.2018
2. Reddit. HomePage, `https://www.redditinc.com/`. Last accessed 12.1.2018
3. tf-idf, `https://en.wikipedia.org/wiki/Tf%E2%80%93idf`