University *of Ljubljana*
Faculty *of Computer and Information Science*

IMapBook collaborative discussions classification

Marko Katrašnik and Erica Zago

Abstract

- First defense (10 points): task selection simple corpusprocessing/analysis
  - Introduction, existing solutions, initial ideas.
- Interim defense (10 points): at least one example of a solution to aproblem
  - Introduction, related work, implemented baseline, future directions
- Final defense (30 points): full submission and presentation
  - Clean Git repository (fully reproducible) and final report

Keywords
Keyword1, Keyword2, Keyword3 ...

Advisors: Slavko Žitnik

Uvod

Namen te seminarske naloge je pripraviti algoritem, ki bo klasificiral sporočila iz zbirke IMapBook. Sporočila so bila napisana v okviru knjižnih krožkov, kjer so sodelovale različne skupine. Vsaka skupina je imela nalogo, da skupaj sestavi odgovor na podano vprašanje, nanašajoče se na preprano knjigo/poglavje. Preko sporočil so se dogovorili kako odgovoriti. Ta sporočila so bila različnih vrst, kot npr. komentarji, diskusije, vprašanja, navodila itd. Naš cilj je klasificirati vsako sporočilo v pripadajočo kategorijo.

Ukvarjala se bova torej s klasifikacijo besedil (angl. text classification). Bolj specifično se bova ukavrjala s klasifikacijo govornih dejanj (angl. speech act classification) in klasifikacijo kratkih besedil (angl. short text classification) [1].

Analiza podatkov

Število sporočil:

- 711 sporočil v prvem zavihku (CREW data)
- 130 sporočil v drugem zabihku (Discussion only)

V tabeli 1 so prikazane kategorije spročil v zavihku CREW data, kjer imajo sporočila tudi pripadajoči skupni končni odgovor. V tabeli 2 pa so prikazane kategorije iz zavihka Discussion, kjer ni pripadajočega skupnega odgovora.

Opazna je zelo neenakomerna porezdelitev števila soročil po kategorijah, kar bo potrebno upoštevati pri uporabi algoritmov, še bolj pa pri uporabi in interpretaciji metrik za evalvacijo.

Na slikah 1 in 2 sta prikazani distribuciji dolžin sporočil (število besed v sporočilu) v zavihkih Crew data in discussion. Distribuciji sta si precej različni, v prvi je veliko več kratkih sporočil, druga je bolj enakomerna, prisotnih je tudi nekaj daljših sporočil. Povprečna dolžina sporočila v CREW data je 11 besed, v discussion pa 42,3 besed.

Metode

Postopek klasifikacije sporočil

- Predprocesiranje in vektoriziranje podatkov
- Izbira modela
- Izbira parametrov, s katerimi bo model delal

Osnovni algoritem

Predprocesiranje podatkov:

- Tekst bo pretvorjen v male tiskane črke
- Če obstajajo bloki razmikov, bi te pretvorili v en razmik
- Tokenizacija - zaporedje besed
- Korenjenje/lematizacija (angl. Stemming/lematization) besed

Table 1. CREW data - Razredi in število sporočil v vsakem razredu.

Kategorija	št. sporočil
assignment instructions	36
assignment instructions question	1
assignment question	10
content discussion	299
content question	18
discussion wrap-up	12
emoticon/non-verbal	12
external material	2
feedback	31
general comment	55
general discussion	9
general question	2
greeting	11
incomplete/typo	10
instruction question	3
logistics	80
observation	2
opening statement	8
outside material	10
response	99

Table 2. Discussion - Razredi in število sporočil v vsakem razredu.

Kategorija	št. sporočil
assignment instructions	1
content discussion	94
content question	5
emoticon/non-verbal	3
feedback	3
general comment	6
incomplete/typo	10
instruction question	1
response	7

Sporočila bi nato pretvorila v N-gram vektorje, podrobneje kombinacijo unigramov in bigramov. Vektorji bi bili seveda numerični in za pridobitev teh bi uporabili TF-IDF metriko. TF-IDF je vrsta meritve, ki ocenjuje, koliko je neka beseda pomembna za dokument v zbirki dokumentov. Izračuna se tako, da se pomnoži dve metriki: kolikokrat se beseda pojavi v dokumentu (angl. Term Frequency - tf) in obratno pogostost dokumenta v nizu dokumentov (angl. Inverse Document Frequency - idf).

Za začetno implementacijo (angl. baseline) bi za klasifikacijo vhodnih tf-idf vektorjev uporabila MLP (angl. Multilayer Perceptron). MLP je klasični tip nevronske mreže. Sestavljen je iz ene ali več plasti nevronov. Podatki so vnešeni v vhodno plast, nato pa gredo skozi eno ali več skritih plasti, ki zagotavljajo ravni abstrakcije. Na izhodni plasti se pa izračunajo napovedi oziroma verjetnosti posameznih razredov.

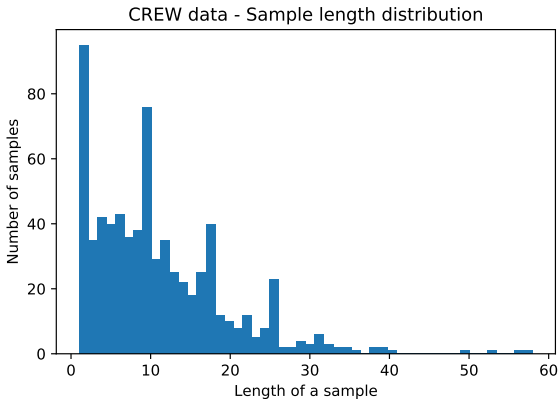

Figure 1. Distribucija dolžine sporočil v zavihku CREW data.

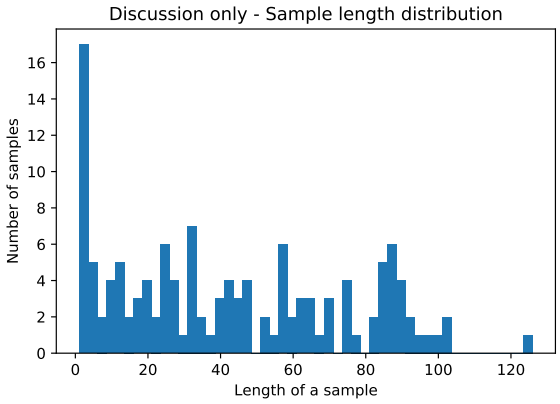

Figure 2. Distribucija dolžine sporočil v zavihku discussion.

MLP je primeren za klasifikacijsko napovedovanje, kjer je vhodnim podatkom dodeljen razred ali oznaka. Delovanje MLP bi lahko primerjala še s kakšnim drugim algoritmom strojnega učenja, na primer z metodo podpornih vektorjev (angl. Support Vector Machine - SVM).

V kasnejšem, naprednejšem algoritmu bi uporabila BERT (angl. Bidirectional Encoder Representations from Transformers) [2], ki je dvosmerni transformator, ki je bil predhodno treniran s kombinacijo modeliranja z zamaskiranim jezikom in napovedovanja naslednjega stavka na velikem korpusu, ki ga sestavljata Toronto Book Corpus in Wikipedia.

Za klasifikacijo besedil se sicer v zadnjem času uporabljajo tudi naprednješi algoritmi, ki upoštevajo tudi vrstni red besed[1]. Za vektorizacijo se uporablja na primer GloVe embedding, za klasifikacijo pa nato konvolucijske (CNN) ali rekurenčne (RNN) nevronske mreže. Težava teh pristopov pa je, da za uspešno učenje nevronskih mrež potrebujemo velike zbirke podatkov, zato takih pristopov zaenkrat nisva mislila uporabiti.

[1] https://developers.google.com/machine-learning/guides/text-classification

Izbira parametrov, s katerimi bo model delal

- Najpomembnejši parameter bo ”Message”, kateri vsebuje vsebino sporočila, kateremu želimo napovedati kategorijo.
- Parameter ”CodePreliminary” je oznaka kategorije in se bo uporabljal za učenje algoritmov in za evalvacijo.
- ”Collab Response” in ”Book ID” (povezano z besedilom knjige) bosta morda uporabna z uporabo metod za računanje podobnosti s sporočilom (Podrobneje predstavljeno v naslednjem razdelku).
- ”isAnswer” in ”Page” lahko zanemarimo, kot lahko tudi ”Topic” (ker vprašanje ni zelo specifično)

Uporaba besedila knjig in skupnega končnega odgovora

Želiva ugotoviti tudi, ali ima uporaba podatkov besedila knjig in skupnega končnega odgovora pozitiven vpliv na uspešnost napovedovanja razredov posameznih sporočil. Glede na to, da ena podmnožica sporočil (discussion) nima podanega podatka o končnem odgovoru, ima pa precej daljša besedila sporočil, bova lahko primerjala tudi koristnost teh dveh lastnosti.

Besedila knjig in skupne končne odgovore bi uporabila tako, da bi izračunala podobnost s sporočilom. Predvidevava namreč, da so si sporočila, v katerih so govorili o vsebini (npr. content discussion) bolj podobna besedilu knjige in končnemu odgovoru. Tako kot pri osnovni klasifikaciji spročil, bi tudi tukaj najprej za vektorizacijo besedil uporabila TF-IDF, kasneje pa bi ga nadomestila z word2vec. Podobnost bi računala s kosinusno podobnostjo.

Podatka o podobnosti z besedilom knjige in končnim odgovorom, bi nato skupaj z napovedjo razreda sporočila (ta napoved bi bila pridobljena z zgoraj opisanimi postopki, ki uporabljajo samo vsebno sporočila) poslala na vhod še enega algoritma strojnega učenja. Uporabila bi random forest, kasneje pa morda še MLP.

Okvirna shema algoritma klasifikacije sporočil z uporabo besedil knjig in končnih skupnih odgovorov je prikazana na sliki 3 .

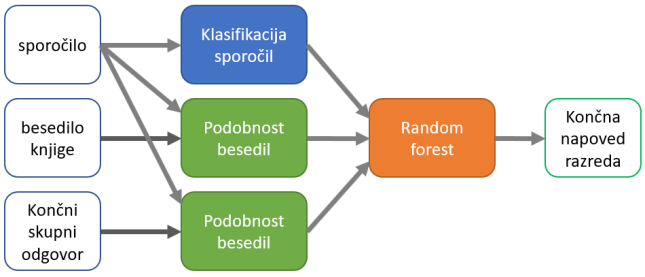

Figure 3. Okvirna shema uporabe besedila knjig in končnih odgovorov.

Drugo

Ideje kaj lahko dodatno analiziramo in kako:

- Na podlagi prejšnjega komentarja (npr. če je to vprašanje) lahko predpostavimo da bo naslednji komentar odgovor
- lahko uporabimo Pseudonym, da vemo kdo govori in če je ta postavil set sporočil (en komentar za drugim) - da vemo če je več zaporednih sporočil pisala ena sama oseba.
- Uporaba slovarjev - npr. če vemo da je omejeno na neke znake ali na neke besede (npr. emoticons, greeting..., verjetno če je na koncu vprašaj bo vprašanje...)
- Ker je dosti kategorij, jih bomo nekaj združili v nadkategorije, za lažjo klasifikacijo
- Dobili bi lahko ključno besedo za vsako kategorijo (npr. ”Hi” za Greeting)
- Mogoče uporabni podatki: Message Time (za predhodno sporočilo)

Testiranje napovedovanja:

- cross validation
- set podatkov razdelimo na dva dela (train in test set) v razmerju 80% - 20% (prvi tab: 568 - 143; drugi tab: 104 - 26)
- train set se še enkrat razdeli (v train in validation set) v razmerjo 80% - 20% (prvi tab: 454 - 114; drugi tab: 83 - 21)
  https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

Ocenjevanje izvedene metode:

- precision and recall
- Mera F (poroča o uspešnosti):
  - utežena
  - mikro in makro
  - ker imamo več razredov sproti poročamo povprečje

  Koristni linki (Področja/sorodna dela - Existing solutions)

  - Speech act analysis in short text classification
  - Natural Language Processing for Pytorch and TensorFlow: https://huggingface.co/transformers/
  - Text classification Google Developers: https://developers.google.com/machine-learning/guides/text-classification/step-2-5
  - Classifying responses on online discussion forums: https://cs224d.stanford.edu/reports/AbajianAaron.pdf
  - Classification of Online Discussions Via Content and Participation (str 820-828): https://link.springer.com/content/pdf/10.1007%2F11881223.pdf
  - Hierarchical speech-act classification for discourse analysis: https://www.sciencedirect.com/science/article/abs/pii/S0167865513000950

- Automated Speech Act Classification in Arabic: http://www.cs.memphis.edu/~vrus/publications/2010/Arabic-SAC.LubnaRusGraesser.pdf
- A Speech Act Classifier for Persian Texts and its Application in Identifying Rumors: https://arxiv.org/ftp/arxiv/papers/1901/1901.03904.pdf
- Tweet Acts: A Speech Act Classifier for Twitter: https://arxiv.org/pdf/1605.05156.pdf
- Short Text Classification: A Survey (str 635-643) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.3331&rep=rep1&type=pdf#page=19

References

[1] Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. Short text classification: A survey. *Journal of multimedia*, 9(5):635, 2014.

[2] Raymond Cheng. Bert text classification using pytorch, Jul 2020.