



# Klasifikacija kratkih sporočil - IMapBook

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 a problem
  - Introduction, related work, implemented baseline, future directions
- Final defense (30 points): full submission and presentation
  - Clean Git repository (fully reproducible) and final report
- Komentr asistenta:
  - zelo jasen opis
  - opis korpusa bo treba še razširita in ga obravnavati v ločenem poglavju, saj ni še nikjer opisan. Opisati ga je potrebno tudi iz različnih vidikov
  - povedal na zagovoru:
    - \* izluščiti še več vidikov: knjige, razredi, skupine

## Keywords

Keyword1, Keyword2, Keyword3 ...

Advisors: Slavko Žitnik

## 1. Uvod

Klasifikacija besedil je podpodročje obdelave naravnega jezika in je proces razvrščanja besedil v organizirane skupine. 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 prebrano knjigo/poglavje. Preko sporočil so se dogovorili kako odgovoriti.

Ukvarjala sva se torej s klasifikacijo besedil (angl. text classification), bolj specifično s klasifikacijo govornih dejanj (angl. speech act classification) in klasifikacijo kratkih besedil (angl. short text classification) [1]. Za klasifikacijo kratkih besedil je značilno, da je klasifikacija nekoliko težja zaradi majhnega števila besed v posameznem sporočilu. Po drugi strani pa so pogosto na voljo dodatni podatki, kot na primer časovno sosledje sporočil in avtor posameznega sporočila. V zbirki IMapBook so poleg tega na voljo še besedila skupnih

končnih odgovorov in vsebine knjig na katere se nanašajo sporočila, zato sva se odločila, da se osredotočiva na uporabo teh podatkov, ki so posebnost uporabljene zbirke.

V okviru drugega dela seminarske naloge sva dopolnila opis zbirke podatkov, ki je predstavljena v razdelku 3.1 in pripravila izhodiščni klasifikator, ter klasifikator, ki uporabi kombinacijo izhoda izhodiščnega algoritma in podobnosti sporočila s končnim odgovorom in vsebino knjige. Izbira parametrov in rezultati so predstavljeni v razdelku 4. V razdelku 5 je predstavljen plan za nadaljnje delo.

## 2. Metode (ideje)

### 2.1 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

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. 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 naprednejši algoritmi, ki upoštevajo tudi vrstni red besed<sup>1</sup>. 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 nevronske mreže potrebujemo velike zbirke podatkov, zato takih pristopov zaenkrat nisva mislila uporabiti.

## 2.2 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 uporabljena za računanje podobnosti s sporočilom (Podrobneje predstavljeno v naslednjem razdelku).
- "isAnswer" in "Page" lahko zanemarimo, kot lahko tudi "Topic" (ker vprašanje ni zelo specifično)

<sup>1</sup><https://developers.google.com/machine-learning/guides/text-classification>

## 2.3 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 sporočil, 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 1.

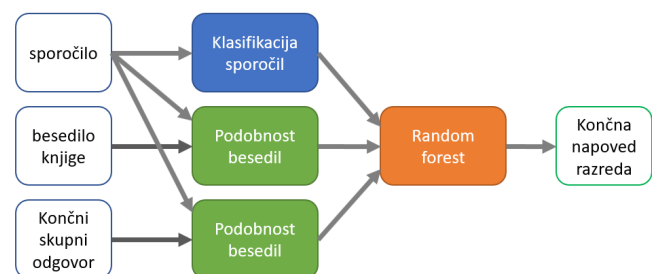


Figure 1. Okvirna shema uporabe besedila knjig in končnih odgovorov.

## 3. Eksperimenti

### 3.1 Zbirka podatkov

Za razvoj algoritmov, učenje modelov in testiranje sva uporabila zbirko podatkov IMapBook. Zbirka je nasatela v okviru knjižnih krožkov, pri katerih so učenci najprej prebrali določeno knjigo, nato pa odgovarjali na zastavljena vprašanja. Odgovore so oblikovali s pomočjo skupinske diskusije, ki je potekala v obliki pošiljanja kratkih sporočil v namenski aplikaciji.

Zbirka je razdeljena na dva dela. V prvem (CREW data) so podana sporočila, ki so si jih učenci pošiljali in tudi končni odgovor, ki so ga napisali kot skupina. V drugem (Discussion only data) so podana le sporočila, brez končnih skupinskih odgovorov. V obeh delih je podano tudi na katero knjigo se nanaša diskusija, v zbirki pa so tudi besedila teh knjig.

### 3.1.1 Obdelava zbirke podatkov

Zbirka podatkov IMapBook ni bila še prečiščena, zato sva podatke pregledala in naredila nekaj sprememb.

Nekaj sporočil je imelo oznake razredov, ki niso bile pričakovane glede na opise razredov, ki so podani v ločeni datoteki v zbirki podatkov. Glede na to, da je bilo v vsakem od takih razredov zelo malo sporočil in ker so bili po vsebini podobni sporočilom v predvidenih razredih, sva jih pridružila k razredom, ki so bili glede na vsebino sporočil najbolj primerni. H katerim razredom sva pridružila sporočila iz posameznega nepričakovanega razreda je prikazano v tabeli 1.

**Table 1.** V tabeli je prikazano h katerim razredom so bile pridružene nepričakovane oznake razredov.

Izvorni razred	Dodano k razredu
assignment instructions question	assignment instruction
general discussion	general comment
observation	general comment
outside material	external material

Pri štirih sporočilih je manjkala oznaka id-ja, ki sporoča na katero knjigo se nanaša diskusija. Glede na to, da so sporočila očitno pripadala k sporočilom pred in za takimi sporočili, sva dodala enak id knjige kot je bil določen v okolici.

V treh primerih skupina ni podala končnega odgovora. Oznake v teh primerih niso bile usklajene, zato sva pri vseh vnesla <Nothing>, kot je bilo zapisano pri enem od treh primerov brez končnega odgovora.

V zbirki podatkov je bilo nekaj razredov z zelo malo pripadajočimi sporočili. Ta težava je še posebej izrazita v delu zbirke discussion. S testiranjem delovanja in v želji, da bi v obeh delih zbirke podatkov uporabila enako pravilo za odstranjevanje razredov z malo primeri, sva določila, da so odstranjena sporočila iz razredov z manj kot petimi primeri. Odstranjeni razredi so označeni v tabelah 2 in 3.

V dveh primerih je manjkal podatek o avtorju sporočila, oziroma njegov psevdonim (Pseudonym). Na ti dve mesti sva napisala besedo "blank".

### 3.1.2 Analiza podatkov

Prečiščene podatke sva nato ponovno analizirala, da bi dobila boljši vpogled v zbirko in na tej podlagi lažje določila nadaljne korake.

Število sporočil:

- 705 sporočil v prvem zavihku (CREW data) - 711 pred odstranjevanjem razredov
- 122 sporočil v drugem zavihku (Discussion only) - 130 pred odstranjevanjem razredov

V tabeli 2 so prikazani razredi sporočil v zavihku CREW data, kjer imajo sporočila tudi pripadajoči skupni končni odgovor. V tabeli 3 pa so prikazani razredi iz zavihka Discussion, kjer ni pripadajočega skupnega odgovora. Pri obeh je podano tudi število sporočil, ki pripadajo posameznemu razredu.

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

**Table 2.** CREW data - Razredi in število sporočil v vsakem razredu.

Razred	št. sporočil
assignment instructions	35
assignment question	10
content discussion	299
content question	18
discussion wrap-up	12
emoticon/non-verbal	13
external material	13
feedback	31
general comment	66
general question (odstranjeno)	2
greeting	11
incomplete/typo	10
instruction question (odstranjeno)	4
logistics	80
opening statement	8
response	99

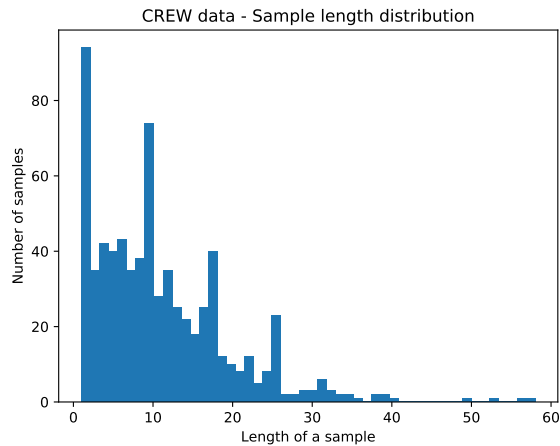
**Table 3.** Discussion - Razredi in število sporočil v vsakem razredu.

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

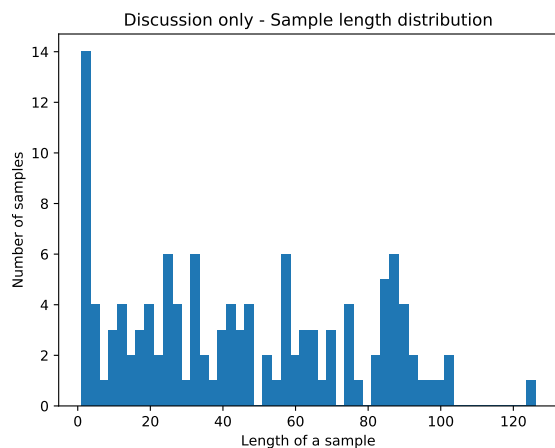
Na slikah 2 in 3 sta prikazani distribuciji dolžin sporočil (število besed v sporočilu) v zavihkih Crew data in discussion. Distribuciji sta si precej različni, v prvi je veliko več kratkih sporočil, druga je bolj enakomerna, prisotnih je tudi nekaj daljših sporočil. Povprečna dolžina sporočila v CREW data je 11 besed, v discussion pa 44,5 besed.

Analizirala sva tudi kateri učenci so bili najbolj aktivni. Aktivnost študenta je bila določena na podlagi števila sporočil, ki jih je poslal. Za CREW data so bili najbolj aktivni: edf-16 s 104 sporočili, edf-15 s 104, pim-11 s 54, pim-12 s 44 in pim-04 s 39 sporočili. Za Discussion only pa: pim-30 s 10 sporočili, dig-11 s 8, pim-05 s 5, pim-14 s 5 in pim-07 s 5 sporočili.

Poleg tega sva analizirala tudi, kdo je največ prispeval h končnemu odgovoru, oziroma sporočila katerega učenca so si



**Figure 2.** Distribucija dolžine sporočil v zavihku CREW data.



**Figure 3.** Distribucija dolžine sporočil v zavihku discussion.

povprečno najbolj podobna s končnim odgovorom. Rezultati so prikazani v tabeli 4. Podobnosti sva računala z uporabo kosinusne podobnosti med tf-idf vektorji sporočil in končnimi odgovori.

Po istem principu je možno ugotoviti kdo je pisal sporočila, ki so bila najbolj podobna vsebini knjige. Za knjigo "The Lady or the Tiger" je oseba s pseudonimom pim-02 dosegla podobnost 0,23024, s knjigo "Just Have Less" so bila najbolj podobna sporočila učenca s pseudonimom dig-04 s podobnostjo 0,25960, za knjigo "Design for the Future When the Future Is Bleak" pa je oseba s pseudonimom dig-14 dosegla podobnost 0,23120.

### 3.1.3 Podobnosti vsebine sporočil s končnimi odgovori in vsebinami knjig

Glede na to, da je eden od namenov te seminarske naloge, da ugotoviva, ali podatki o končnem odgovoru in vsebini knjige na katero se nanašajo sporočila pripomorejo k uspešnosti napovedovanja razredov sporočil, sva preverila kakšna je povprečna podobnost med sporočili posameznega razreda

**Table 4.** Kdo je največ prispeval h končnemu odgovoru

Vprašanje	Pseudonim	Podobnost
1.1	dig-01	0,26076
1.2	pim-06	0,15764
1.3	dig-04	0,0
1.4	dig-08	0,08012
1.5	dig-10	0,25847
2.1	dig-13	0,13365
2.2	dig-14	0,40414
2.3	dig-16	0,15382
2.4	dig-19	0,0
2.5	dig-24	0,16821
3.1	pim-02	0,25516
3.2	blank	0,20895
3.3	pim-09	0,31015
3.4	pim-11	0,12908
3.5	pim-14	0,31100
4.1	pim-16	0,0
4.2	pim-20	0,05845
4.3	pim-23	0,44649
4.4	pim-25	0,05385
5.1	edf-01	0,15841
5.2	edf-08	0,34438
5.3	edf-05	0,27309
6.1	edf-10	0,07543
6.2	edf-12	0,22879
6.3	edf-15	0,10225

in končnimi odgovori, ter vsebinami knjig.

Tudi na tem mestu sva podobnost računala z uporabo kosinusne podobnosti med tf-idf vektorji sporočil in končnimi odgovori, ter vsebinami knjig.

Ugotovila sva, da se povprečne podobnosti med razredi precej razlikujejo, iz česar sklepava, da obstaja dobra podlaga za izboljšanje uspešnosti razreda sporočil. Opazila sva tudi, da so si vrednosti kosinusne podobnosti s končnim odgovorom in vsebino knjige zelo podobne in so višje pri razredih povezanih z vsebino knjige. Povprečne podobnosti so prikazane v tabelah 5 in 6.

### 3.2 scikit-learn

Za implementacijo izhodiščne rešitve (angl. baseline) sva uporabila knjižnico scikit-learn. Uporabila sva StratifiedShuffleSplit za delitev podatkov na učno in testno množico (70% učna in 30% testna množica), TfidfVectorizer za generiranje TF-IDF vektorjev, MLPClassifier za napovedovanje razreda z MLP in metrics za evalvacijo.

Za upoštevanje podobnosti s končnimi odgovori in vsebinami knjig sva prav tako uporabila knjižnico scikit-learn. Uporabila sva funkcijo cosine\_similarity za računanje podobnosti in RandomForestClassifier za pridobivanje končne napovedi razredov. Preizkusila sva dva načina združevanja podatkov. Pri prvem sva združila napoved razreda, ki ga vrne MLP izhodiščnega algoritma in kosinusno podobnost. Tako sva

**Table 5.** Kosinusne podobnosti med sporočili iz zavihka CREW, končnimi odgovori in besedili knjig.

Razred	Končni odgovor	Knjiga
assignment instructions	0,10	0,15
assignment question	0,05	0,04
content discussion	0,20	0,19
content question	0,19	0,15
discussion wrap-up	0,05	0,08
emoticon/non-verbal	0,00	0,00
external material	0,11	0,10
feedback	0,08	0,07
general comment	0,08	0,09
greeting	0,01	0,01
incomplete/typo	0,00	0,00
logistics	0,08	0,10
opening statement	0,05	0,07
response	0,03	0,04

**Table 6.** Kosinusne podobnosti med sporočili iz zavihka Discussion in besedili knjig.

Razred	Knjiga
content discussion	0,31
content question	0,17
general comment	0,13
incomplete/typo	0,00
response	0,02

imela le dve ali tri vhodne značilke za končni klasifikator z naključnim gozdom. Pri drugem načinu pa sva združila vektor z verjetnostmi posameznih razredov in kosinusne podobnosti.

## 4. Rezultati

Zaenkrat sva testirala delovanje izhodiščnega algoritma, ki iz sporočil tvori TF-IDF vektorje in razred napoveduje z uporabo MLP-ja, ter vpliv uporabe podatkov končnega skupinskega odgovora in vsebine knjige.

Parametre izhodiščnega algoritma sva nastavila tako, da sva algoritem iterativno izboljševala s spreminjanjem posameznih parametrov. Skozi ta postopek sva ugotovila, da korenjenje in lematizacija nimata pozitivnega učinka na uspešnost napovedovanja, enako velja tudi za odstranjevanje stop besed (angl. stop words). Razlog za to je morda, da so sporočila kratka in je zato pomembna vsaka informacija, ki jo lahko uporabimo, medtem, ko je pri daljših besedilih pomembneje prodobiti bistvo brez podrobnosti.

Čeprav sva predvidela uporabo kombinacije unigramov in bigramov, sva ugotovila, da je delovanje boljše če se uporabijo le unigrami.

Zmanjševanje dolžine TF-IDF vektorjev, oziroma zmanjševanje nabora značilk se ni izkazalo za učinkovito, je pa manjše izboljšanje prineslo pravilo, da se mora beseda pojaviti vsaj v dveh sporočilih, da se doda kot značilka.

Pri MLP-ju sva uporabila optimizator adam, nevronska mreža ima en sam skriti nivo z 20 nevroni.

V tabeli 7 so prikazani rezultati evalvacije izhodiščnega algoritma. Uporabila sva metrike: točnost (angl. accuracy), makro preciznost (angl. precision), makro priklic (angl. recall) in makro mero F1.

**Table 7.** Rezultati evalvacije izhodiščnega algoritma

	CREW	Discussion
accuracy	0,61	0,89
macro precision	0,52	0,53
macro recall	0,46	0,60
macro F1	0,45	0,56

### 4.1 Uporaba podobnosti

Ugotovila sva, da je napovedovanje razredov, če je združena le končna napoved razreda in kosinusna podobnost, bolj uspešno, kot če je uporabljen celoten vektor verjetnosti posameznih razredov.

Pri napovedovanju razredov sporočilom iz dela zbirke CREW z uporabo podobnosti zaenkrat nisva dosegla izboljšanja, opazno je bilo celo poslabšanje. Vrednost makro mere F1 se je iz 0,446 pri dodanem podatku o podobnosti s končnim odgovorom znižala na 0,395, pri dodani podobnosti z vsebino knjige pa celo na 0,361. Ko sta bili dodani obe podobnosti, je bila vrednost mere 0,375.

Sva pa dosegla izboljšanje pri uporabi podobnosti z vsebino knjige pri delu podatkov Discussion. Vrednost makro mere F1 se je iz 0,561 zvišala na 0,723. Razlog za izboljšanje v tem primeru je morda to, da so v tem delu sporočila daljša in je zato podobnost z drugimi daljšimi besedili bolj smiselna.

**Table 8.** Rezultati evalvacije algoritma, ki upošteva podobnosti s končnimi odgovori in vsebinami knjig

	accuracy	F1	accuracy	f1
končni odgovor	0,608	0,395	/	/
vsebina knjige	0,594	0,361	0,919	0,723
oboje	0,590	0,375	/	/

## 5. Plan za nadaljnje delo

V nadaljevanju se bova posvetila klasifikaciji sporočil s pomočjo globokega učenja. Uporabila bova BERT s pomočjo knjižnice Transformers. Najprej bova poskusila kakšnega od modelov, ki so na voljo v okviru knjižnice Transformers. Nato pa bova model (arhitekturo) poskusila prilagoditi za uporabo s kratkimi besedili. Za prilagoditev bova uporabila ideje iz literature.

Metode globokega učenja bova uporabila tudi za vektorizacijo besedil (vektorske vložitve besed) pred računanjem podobnosti med sporočili in končnimi odgovori, ter vsebinami knjig. V začetnem planu sva predvidela uporabo word2vec,

vendar glede na to, da bova uporabila BERT že pri začetni klasifikaciji sporočil, bo verjetno smiselno preveriti delovanje metode BERT še na tem mestu.

## 6. 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 razmerju 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 (poročila o uspešnosti):
  - utežena
  - mikro in makro

– ker imamo več razredov sproti poročamo povprečje

Sorodna dela:

- 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>
- BERT Text Classification Using Pytorch: <https://towardsdatascience.com/bert-text-classification-using-pytorch-723dfb8b6b5b>

## References

- <sup>[1]</sup> Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. Short text classification: A survey. *Journal of multimedia*, 9(5):635, 2014.
- <sup>[2]</sup> Raymond Cheng. Bert text classification using pytorch, Jul 2020.