ONJ - seminar 2

Žiga Simončič
Fakulteta za računalništvo in informatiko
Univerza v Ljubljani
Ljubljana, Slovenija
zs3179@student.uni-lj.si

Klemen Randl
Fakulteta za računalništvo in informatiko
Univerza v Ljubljani
Ljubljana, Slovenija
kr3037@student.uni-lj.si

***Povzetek*—V dokumentu predstavimo našo rešitev problema ocenjevanja kratkih odgovorov na podlagi izhodiščnega teksta in manjšega nabora že ocenjenih odgovorov na podano vprašanje. Predstavimo tri tipe modelov (A, B, C). Model A ocenjuje samo na podlagi izhodiščnega teksta in enega pravilnega odgovora. Modelu B dodamo večji nabor pravilnih in nepravilnih odgovorov, v model C pa vpeljemo še zunanji vir. Za modela A in B smo razvili po eno verzijo, za model C pa smo razvili tri. Razvijamo v sodelovanju z IMapBook [1]. Izvorna koda je dostopna na portalu Github [7].**

***Index Terms*—ocenjevanje kratkih odgovorov, semantična podobnost**

I. Uvod

Ocenjevanje (kratkih) odgovorov na vprašanja je zanimiv problem, ki ima uporabo na več področjih. En izmed takih primerov uporabe je v domeni IMapBook. IMapBook je spletna platforma, kjer lahko učenci skozi igro berejo kratke zgodbice in se tako na zabaven način uvajajo v svet književnosti in literature. Omenjene igre obsegajo različne stvari v povezavi s prebrano zgodbo. Ena izmed iger je tudi odgovarjanje na kratka vprašanja, kjer ima učenec prost vnos besedila, odgovor na vprašanje pa je skrit v tekstu oz. prebrani zgodbi. Ker je teh učencev veliko, si želimo postopek ocenjevanja odgovorov avtomatizirati. Skupno imamo na voljo nekaj sto primerov vprašanje-odgovor (učna množica).

V kontekstu te naloge bomo zgradili tri modele: A, B in C. Model A bo zgrajen na podlagi le enega pravilnega odgovora in izhodiščnega besedila. Pri modelu B bomo uporabili cel nabor učne množice, modelu C pa bomo poleg tega dodali še zunanji vir podatkov.

II. Pregled področja

Našli smo raziskovalno nalogo, katere tematika je zelo podobna IMapBook problemu. Naslov članka je "Automatic Natural Language Processing and the Detection of Reading Skills and Reading Comprehension" [2]. V članku je govora o dveh pristopih, s katerimi lahko z uporabo obdelave naravnega jezika ocenimo bralno razumevanje. V članku je omenjeno orodje "Reading Strategy Assessment Tool"ali krajše R-SAT, ki je precej podobno IMapBook aplikaciji, saj uporabniku po delih prikazuje besedilo, nato pa na določenih točkah bralcu postavlja vprašanja, ki se navezujejo na prebrano besedilo. Vprašanja so lahko direktna ali indirektna. Indirektna vprašanja so bolj odprtega tipa in uporabnika povprašajo po njegovih občutkih ali mnenju o prebranem, direktna vprašanja pa so veliko bolj specifična in so podobnega tipa kot so vprašanja v našem IMapBook problemu. Na podlagi odgovorov na ta vprašanja se nato oceni bralčevo razumevanje prebranega besedila. V omenjenem članku sta bila uporabljena dva algoritma iz področja obdelave naravnega jezika in sicer Latentna semantična analiza (LSA) ter ujemanje besed (polno ujemanje ter Soundex ujemanje). LSA z uporabo statističnega modela oceni semantično podobnost med dvema odgovoroma, ujemanje besed pa je veliko bolj preprost algoritem, ki samo preveri ali se uporabljajo semantično podobne besede v odgovoru in na podlagi tega oceni podobnost. Tak pristop je sicer računsko zelo enostavnejši, ampak je tudi manj sofisticiran. Majhno izboljšavo sicer prinese Soundex ujemanje, ki upošteva tudi morebitne slovnične napake bralca pri pisanju odgovora.

Našli smo tudi veliko ostalih člankov s področja avtomatskega ocenjevanja kratkih odgovorov. V članku z naslovom "Fast and Easy Short Answer Grading with High Accuracy" [3] se podani odgovor primerja z referenčnim tako, da ugotovi koliko semantično podobnih besed je v obeh odgovorih. Izračuna se podobnost semantičnih vektorjev z uporabo kosinusne podobnosti. V članku se pred računanjem podobnosti med podanim ter referenčnim odgovorom iz obeh odstranijo vse besede, ki se pojavijo v vprašanju. Razlog za to naj bi bil to, da se podanemu odgovoru ne poveča ocena, če je veliko besed ponovljenih iz vprašanja. Uporabljena je tudi kontekstualna različica TF-IDF metrike za uteževanje besed. Torej TF-IDF se izračuna na podlagi konteksta (domene) vprašanja ter odgovorov tako, da se za pomembne besede v vprašanju ter referenčnem odgovoru preišče 10 najbolj primernih zadetkov na Wikipediji. In nato se TF besede v podanem odgovoru izračuna na podlagi tega, kolikokrat se pojavi v zadetkih iz relevantnega področja Wikipedije. IDF posamezne besede pa se izračuna na podlagi zadetkov iz Wikipedije za 25 različnih domen (Matematika, Šport, Politika, itd). Še ena zanimiva metrika je razmerje med številom besed v podanem odgovoru in številom besed v refernečnem odgovoru. Če je to razmerje prenizko pomeni, da podani odgovor ni dovolj specifičen oziroma da je prekratek.

Še en zanimiv članek ki si postavlja podobna vprašanja je "Text-to-text Semantic Similarity for Automatic Short Answer Grading" [4]:

- Kako najbolje izmeriti semantično podobnost za potrebe ocenjevanja odgovorov na kratka vprašanja?

- Kako domena vprašanj ter podatki, na katerih učimo algoritem vplivajo na natančnost ocene?
- Ali lahko uporabimo najbolje ocenjene podane odgovore za izboljšanje natančnosti meritve semantične podobnosti?

V tem članku je opisanih več pristopov za pridobitev čim bolj natančne ocene pravilnosti odgovora na dano kratko vprašanje. Delijo se na metrike na podlagi znanja (angl. Knowledge-Based Measures) ter metrike na podlagi korpusa (angl. Corpus-Based Measures). Med rezultati v omenjenem članku lahko vidimo, da načeloma slednji dajejo boljše rezultate. Med njih spada že prej omenjena LSA, tako kot tudi Eksplicitna semantična analiza (ESA).

Članek "Corpus-based and Knowledge-based Measures of Text Semantic Similarity" [6] nam je dal zanimivo idejo za model C, kjer se z uporabo Wordnet-a izračuna semantično podobnost dveh stavkov. Članek opisuje kako lahko z uporabo Wordnet-a primerjamo označene besede dveh stavkov ter ugotovimo do kakšne mere sta si stavka podobna. Če sta v obeh stavkih besedi, ki imata enak pomen glede na Wordnet, potem se smatra da sta stavka vsaj malce semantično podobna. Več kot je takih besed, bolj podobna sta si stavka. To smo uporabili ko smo primerjali podani odgovor z odgovori iz učne množice, da smo ugotovili ali je podani odgovor v splošnem semantično podoben odgovorom, ki so ocenjeni z visoko oceno ali ne.

III. Metode

V skladu z navodili naloge smo razvili tri modele za ocenjevanje pravilnosti odgovorov. Model A, kjer smo za napovedovanje ocene podanega odgovora uporabili zgolj referenčni tekst ter en pravilen odgovor. Nato smo učinkovitost napovedovanja izboljšali z modelom B, kjer smo poleg relevantnega teksta uporabili večjo množico odgovorov, katerim je bila ročno pripisana ocena. Odgovor je bil lahko bodisi pravilen (ocena 1), delno pravilen (ocena 0.5) ali nepravilen (ocena 0). Model B smo nadgradili še z modelom C, kjer smo uporabili še zunanje vire, kot je recimo Wordnet, da nam je pomagal pri boljšem napovedovanju pravilnosti podanega odgovora.

A. Model A

Pri modelu A smo sprva pravilnost odgvorov napovedovali s pomočjo TF-IDF metrike. To smo naredili tako, da smo za posamezno vprašanje referenčni odgovor ter relevantni tekst najprej tokenizirali ter lematizirali in nato pridobili TF-IDF. Nato smo na podlagi TF-IDF vektorja izračunali kosinusno podobnost s podanim odgovorom, ki je bil prav tko tokeniziran ter lematiziran. Tako smo na podlagi izračunane kosinusne podobnosti lahko napovedali, ali je podani odgovor napačen, delno pravilen ali v celoti pravilen glede na to, do kolikšne mere se je ujemal z referenčnim odgovorom.

V postopek smo vpeljali tudi orodje CoreNLP [5], s pomočjo katerega smo lahko iz referenčnega odgovora ter relevantnega teksta izluščili semantične (relacijske) trojčke, ki smo jih nato primerjali z semantičnimi trojčki podanega odgovora ter na podlagi tega poizkušali napovedati pravilnost odgovora. Pri tem smo uporabili tudi zaznavanje koreferenčnosti. V primerih kjer iz kratkih odgovorov ne moremo pridobiti uporabnih semantičnih trojčkov, pa smo uporabimo prej omenjeno kosinusno podobnost.

Iz člankov, ki smo jih pridobili pri pregledu področja smo v model A dodali še dva pristopa, za katera smo upali da bosta pripomogla k natančnosti napovedi. Z enim pristopom smo vse besede, ki se pojavijo v vprašanju odstranili iz obeh odgovorov (referenčnega ter tistega, ki ga ocenjujemo). S tem smo želeli preprečiti, da je podani odgovor dobil visoko oceno samo zato, ker so bile v njem uporabljene enake besede kot v vprašanju. To smo seveda storili pred računanjem TF-IDF metrike ter pred razbitjem odgovorov na semantične trojčke. Drug pristop pa je upošteval dolžino podanega odgovora v primerjavi z referenčnim. Če je bil pravilen referenčni odgovor več kot dvakrat daljši od podanega, smo smatrali da je podani odgovor nepopoln oziroma ne vsebuje vseh potrebnih informacij ter smo mu zato pripisali določen odbitek. Pri takem pristopu je seveda potrebno predpostavljati, da so vsi referenčni odgovori podani kratko in jedrnato ter da vsebujejo samo relevantno informacijo. Za našo učno množico mogoče tega ne bi smeli kar tako predpostavljati, ampak smo se odločili vseeno preizkusiti, kako se tak pristop obnese.

B. Model B

Model B je zelo podoben modelu A, le da smo tukaj pri oceni podanega odgovora uporabili več referenčnih odgovorov z njihovimi pripadajočimi ocenami in ne samo enega. Tukaj smo uporabili samo tiste izmed referenčnih odgovorov, ki so bili vnaprej ocenjeni kot v celoti pravilni (ocena 1). Delno pravilnih ter nepravilnih nismo upoštevali. Podani odgovor smo nato primerjali s pravilnimi odgovori ter uporabili iste pristope kot v modelu A pri napovedovanju ocene odgovora. Pričakovali smo da bo imel model B večjo natančnosti napovedi, saj ima na voljo precej več podatkov kot model A.

C. Model C

Pri modelu C, kjer smo lahko uporabljali dodatne vire, smo razvili tri različice, vsako s svojim pristopom. Pri prvem modelu C (imenujmo ga C1), smo z uporabo Wordneta ocenjevali podobnost odgovorov ter na podlagi tega podajali oceno. Podani odgovor smo oblikoskladenjsko označili ter ga primerjali z vsemi odgovori iz učne množice, ki so bili prav tako oblikoskladenjsko označeni. Tudi tukaj smo uporabili zgolj pravilne odgovore iz učne množice, kot pri modelu B. Odgovore smo med seboj vsebinsko primerjali z uporabo Wordnet-a in sicer smo ocenjevali, kolikšen del referenčnega ter podanega odgovora se vsebinsko ujema (vsebuje iste besede ali sopomenke). Nato smo na podlagi tega ujemanja ocenili podani odgovor kot napačen, če je bilo ujemanje slabo ali ga sploh ni bilo. Če je bilo ujemanje prisotno, ampak ni bilo dovolj za pravilen odgovor, smo odgovor ocenili kot delno pravilen, sicer pa v celoti pravilen. Tudi pri tem modelu smo poizkušali pri primerjavi iz obeh odgovorov odstraniti vse besede, ki se pojavijo v vprašanju, da smo ocenjevali samo

relevantno podobnost med dvema odgovoroma, in ne besede, ki se ponovijo iz vprašanja.

Naš drugi model C (imenujmo ga C2) je osnovan na modelu B, z eno bistveno razliko. Pri modelu B se iz referenčnih odgovorov izluščijo semantični trojčki, ki se nato primerjajo z semantičnimi trojčki, pridobljenimi iz podanega odgovora. Pri modelu C2 pa se semantičnim trojčkom iz učne množice doda še nekaj dodatnih semantičnih trojčkov, ki se jih pridobi s pomočjo Wordnet-a. Besedam iz prvotno pridobljenih trojčkov se torej poišče sopomenke in tudi te sopomenske trojčke se uporabi pri primerjavi z podanim odgovorom. S tem smo želeli doseči da če je bil podani odgovor zapisan drugače kot referenčni (recimo na drugačen način ali pa da so bile uporabljene sopomenke), da bi vseeno zaznali semantično podobnost ter tudi takemu odgovoru podali pravično oceno.

Izdelali smo tudi tretji C model (imenujmo ga C3). Tukaj smo poizkusili nadgraditi naš model C1, tako da smo vpeljali detekcijo najbolj pomembnih besed v odgovoru. To smo naredili tako, da smo v postopku predprocesiranja za vsak odgovor iz učne množice izračunali njegovo podobnost z ostalimi odgovori, ki so bili ocenjeni kot pravilni. Nato smo za ta odgovor odstranjevali besede eno po eno ter preverjali kako to vpliva na podobnost s pravilnimi odgovori. Če se je podobnost zmanjšala, pomeni da smo odstranili besedo, ki je ključna za pravilen odgovor in to besedo zato označimo kot pomembno. Ko smo ta postopek opravili za vse besede vseh referenčnih odgovorov iz učne množice, dobimo za vsako vprašanje seznam besed, ki so ključne za odgovor na to vprašanje. In ko nato ocenjujemo odgovor iz testne množice, ga preprosto primerjamo s tem seznamom pomembnih besed in če je ujemanje dovolj dobro, odgovor ocenimo kot pravilen. Če je ujemanje slabše, pa odgovor ocenimo kot delno pravilen ali nepravilen.

IV. Rezultati ter diskusija

V tem poglavju so predstavljeni rezultati naših modelov ter nad njimi opravljenih testov.

Modelov imamo torej 5 (en A model, en B model in tri C modele), vsakega smo parametrizirali ter poganjali teste za vse kombinacije teh parametrov. Za testiranje smo uporabljali 5-prečno preverjanje povsod, razen pri modelu A, saj tam to ni prišlo v poštev.

Modele smo preverjali prečno tako, da smo učno množico razdelili na petine. Nato smo najprej prvo petino primerov uporabili kot testno množico ter ostale štiri petine kot učno množico. Nato smo kot testno množico uporabili samo drugo petino primerov ter ostalo kot učno. Podobno smo naredili še za tretjo, četrto in peto petino ter na podlagi ter petih setov rezultatov pridobili čim boljši približek dejanske natančnosti našega modela. S tem smo se želeli izogniti temu, da bi naši modeli imeli dobro natančnost napovedovanja samo za specifične testne množice. Hoteli smo čim boljše rezultate v povprečju za celotno testno množico in to smo s prečnim preverjanjem tudi dosegli. Kot mero uspešnosti smo uporabili mikro in makro F1 oceno preverjanja. Parametre z najboljšo povprečno F1 oceno preverjanja za posamezen model smo uporabili pri vključitvi tega modela na strežnik za testiranje. Ker pa imamo tri verzije C modela, smo na strežnik vključili tisto ki je najbolj natančna (seveda z njenimi najbolj optimiziranimi parametri).

#	remove	use_length	Cosine	openie	F1 micro	F1 macro
1	False	False	True	0	73%	42%
2	False	False	True	1	18%	17%
3	False	False	True	2	18%	17%
4	True	False	True	0	72%	49%
5	True	False	True	1	17%	17%
6	True	False	True	2	17%	17%
7	False	True	True	0	71%	49%
8	False	True	True	1	14%	16%
9	False	True	True	2	14%	16%
10	True	True	True	0	65%	43%
11	True	True	True	1	13%	16%
12	True	True	True	2	13%	16%
13	False	False	False	1	11%	6%
14	False	False	False	2	11%	6%

Tabela I
Rezultati za različne parametre modela A

A. Model A

Parametri modela A so naslednji:

- remove (True ali False; uporaba odstranjevanja besed iz vprašanja iz obeh odgovorov)
- use_length (True ali False; upoštevanje dolžino podanega odgovora v primerjavi z referenčnim)
- use_cosine (True ali False; ali se pri napovedi ocene uporabi tudi kosinusna podobnost ali samo podobnost na podlagi semantičnih trojčkov)
- openie (do kaknšne mere je uporabljen CoreNLP openie; 0 - izključen, 1 - vključen, brez zaznavanja koreferenčnosti, 1 - vključen z zaznavanjem koreferenčnosti)

Kot opazimo iz tablele rezultatov za model A (tabela I), je z naskokom najbolj uspešno napovedovanje pravilnosti odgovora brez semantičnih trojčkov, samo z kosinusno podobnostjo. Za nabor parametrov, ki je predstavljen v vrsticah 1, 4, 7 in 10 je F1 meritev izrazito boljša kot pri ostalih. Vidimo da s spreminjanjem ostalih parametrov dobimo lahko malo boljšo mikro F1 meritev, zato pa malce slabšo makro F1 meritev (ali obratno). Vidimo da odstranjevanje besed ali preverjanje dolžine odgovora niti približno nima takega vpliva na natančnost kot jo ima vključevanje semantičnih trojčkov. Predvidevamo lahko da je za model A podanih premalo informacij oziroma podatkov učne množice, da bi se tak pristop izplačal. Na strežnik za testiranje odgovorov je vključena verzija modela A z naborom parametrov, ki so predstavljeni v 4. vrstici.

B. Model B

Parametri modela B so isti kot za model A, le da tu nismo uporabili preverjanja dolžine podanega odgovora. Rezultati so v tabeli II.

Opazimo da tukaj ni več tako nizke učinkovitosti semantičnih trojčkov, predvidoma zato ker je na voljo veliko večja učna množica. Še vedno pa najboljšo mikro F1 meritev

#	remove	Cosine	openie	F1 micro	F1 macro
1	False	True	0	67%	38%
2	True	True	0	62%	40%
3	False	True	1	65%	48%
4	False	True	2	60%	46%
5	True	True	1	50%	42%
6	True	True	2	39%	37%
7	False	False	1	63%	36%
8	False	False	2	58%	35%
9	True	False	1	45%	29%
10	True	False	2	34%	24%

Tabela II
REZULTATI ZA RAZLIČNE PARAMETRE MODELA B

#	remove	F1 micro	F1 macro
1	True	62%	43%
2	False	65%	46%

Tabela III
REZULTATI ZA RAZLIČNE PARAMETRE MODELA C1

doseže zgolj kosinusna podobnost, brez semantičnih trojčkov (vrstica 1). Zanimiva stvar, ki jo opazimo je tudi to, da pri semantičnih trojčkih vključevanje zaznavanja koreferenčnosti vedno poslabša natančnost modela. Pri odstranjevanju besed vidimo, da se pri zgolj kosinusni podobnosti (vrstici 1 in 2) sicer poslabša mikro F1 meritev, se pa zato poveča makro F1 meritev. Odstranjevanje besed pri semantičnih trojčkih se pa ne izkaže za učinkovit pristop, predvidoma zato ker z odstranjevanjem pomembnih besed iz odgovora ne dobimo več logičnih in smiselnih semantičnih trojčkov in se zato natančnost modela opazno poslabša. Na strežnik za testiranje odgovorov je vključena verzija modela B z naborom parametrov, ki so predstavljeni v 3. vrstici.

C. Model C1

Naš prvi model C ima samo en parameter in sicer odstranjevanje besed. Rezultati njegove učinkovitosti so prikazani v tabeli III

Vidimo da se pri modelu C1 odstranjevanje besed ne obrestuje, saj se zniža natančnost ocenjevanja. Tukaj bi mogoče pričakovali da se bo rezultat izboljšal, saj se v tem modelu primerja podobnost odgovorov z Wordnet-om in bi v teoriji odstranjevanje besed, ki se pojavijo v vprašanju pomenilo, da se pri odgovorih primerjajo resnično samo relevantne besede.

D. Model C2

Naš drugi model C ima štiri parametre:

- remove (True ali False; uporaba odstranjevanja besed iz vprašanja iz obeh odgovorov)
- use_cosine (True ali False; ali se pri napovedi ocene uporabi tudi kosinusna podobnost ali samo podobnost na podlagi semantičnih trojčkov)
- openie (do kaknšne mere je uporabljen CoreNLP openie; 0 - izključen, 1 - vključen, brez zaznavanja koreferenčnosti, 1 - vključen z zaznavanjem koreferenčnosti)
- no_of_synonyms (število sinonimov, ki se jih za vsako besedo semantičnega trojčka poišče v Wordnet-u)

#	remove	openie	no_of_synonyms	F1 micro	F1 macro
1	False	1	1	65%	48%
2	True	1	1	50%	42%
3	False	2	1	61%	47%
4	True	2	1	40%	37%
5	False	1	2	66%	48%
6	True	1	2	50%	42%
7	False	2	2	61%	46%
8	True	2	2	40%	37%
9	False	1	3	66%	48%
10	True	1	3	50%	42%
11	False	2	3	61%	46%
12	True	2	3	40%	37%

Tabela IV
REZULTATI ZA RAZLIČNE PARAMETRE MODELA C2

#	importantWord	score05	score10	F1 micro	F1 macro
1	0.01	0.5	0.6	58%	43%
2	0.01	0.4	0.5	71%	52%
3	0.01	0.3	0.4	72%	41%
4	0.02	0.5	0.6	64%	42%
5	0.02	0.4	0.5	71%	49%
6	0.02	0.3	0.4	72%	40%
7	0.03	0.5	0.6	64%	39%
8	0.03	0.4	0.5	70%	45%
9	0.03	0.3	0.4	70%	38%

Tabela V
REZULTATI ZA RAZLIČNE PARAMETRE MODELA C3

Rezultati so prikazani v tabeli IV

Tudi pri modelu C2 opazimo, da se odstranjevanje besed ne obrestuje. Prav tako opazimo da zaznavanje koreferenčnosti poslabša natančnost, kot pri prejšnjih modelih. Število sinonimov ki jih poiščemo za posamezno besedo iz semantičnega trojčka pa nima bistvenega vpliva na natančnost, kot vidimo iz tabele.

E. Model C3

Naš zadnji model ima malce drugačne parametre od ostalih. Parametri so naslednji:

- importantWord (koliko se mora podobnost zmanjšati po odstranitvi besede, da se ta beseda smatra kot pomembna)
- score05 ter score10 (ko odgovor iz testne množice primerjamo z seznamom pomembnih besed, kolikšna je meja da je odgovor ocenjen kot delno pravilen ali v celoti pravilen)

Rezultati zadnjega testiranja so prikazani v tabeli V. Iz teh rezultatov je razvidno, da gre tukaj bolj za drobno kalibriranje mejnih parametrov modela. Najboljše rezultate prinesejo parametri v vrstici 2 in 3, kjer je bolj ohlapen kriterij za pomembnost besede. Ta verzija C modela se je izkazala za najbolj zanesljivo izmed treh, tako da je na strežnik za testiranje odgovorov vključena verzija modela C z naborom parametrov, ki so za to implementacijo predstavljeni v 2. vrstici.

V. ZAKLJUČEK

S pomočjo 5-prečnega testiranja smo ugotovili, da se tudi precej preproste metode, kot je recimo kosinusna podob-

Model	F1 mikro	F1 macro
A	72%	49%
B	65%	48%
C	71%	52%

Tabela VI
NATANČNOSTI NAŠIH MODELOV

nost, precej dobro obnesejo. Na splošno vpeljava semantičnih trojčkov ni prinesla želenih rezultatov, še posebej če smo upoštevali tudi koreferenčnost, saj je to natančnost naših modelov konsistentno poslabšalo. Ugotovili smo tudi, da odstranjevanje besed, ki nastopajo v vprašanju iz obeh odgovorov, ne pomaga kaj veliko, razen v redkih primerih. Preverjanje dolžine odgovora se tudi izkaže za več ali manj neuporabno.

Pri soočanju s problematiko tega problema smo opazili, da je veliko odgovorov iz podane učne množice ocenjenih kot popolnoma pravilnih. Takih je veliko več kot pa odgovorov ocenjenih kot delno pravilnih ali nepravilnih. Učna množica je malce neuravnovešena, tako da je bilo malce težje naše modele optimizirati da so pravilno napovedovali delno pravilne odgovore, recimo.

Odgovore v učni množici sta ocenjevala dva, Dr. Glenn Smith in Amber. Glede na Cohenov koeficient kapa za statistiko sta se ocenjevalca strinjala 71 procentno, kar je precej dobro ujemanje. Dr. Smith je dosegel 87 procentno ujemanje s končno oceno, Amper pa malo manjše in sicer 82 procentno.

Še ena stvar ki jo lahko izpostavimo je glede semantičnih trojčkov. Tekom razvijanja ter testiranja naših modelov, ki uporabljajo Stanford CoreNLP orodje smo opazili da so ti trojčki velikokrat nelogični ali nejasni, tako da je njihova uporaba ter obdelava malce težja kot smo sprva pričakovali in zato njihov učinek mogoče ni tak kot bi si želeli. Rezultati to tudi odražajo.

Izmed naših treh modelov se najbolj presenetljivo dobro obnese model A. Primerja se z modelom C3, ki ima neprimerljivo več informacij ter virov na voljo. Razlogov za to je lahko več. Mogoče bi bilo potrebno bolj pazljivo nastaviti robne parametre modela, mogoče na kak drugačen način uporabiti podatke iz učnega modela, mogoče pa še preučiti ter izpolniti kakšen drugačen pristop k reševanju tega problema. Vsekakor je vse to razlog za morebitno nadaljnje raziskovalno delo na tem področju.

LITERATURA

[1] Dr. Glenn Smith (2018). Imapbook. Povzeto 18.12.2018, iz https://www.imapbook.com/

[2] Boonthum, Chutima & Mccarthy, Philip & Lamkin, Travis & Jackson, G & Magliano, Joe & McNamara, Danielle. (2011). Automatic Natural Language Processing and the Detection of Reading Skills and Reading Comprehension.

[3] Sultan, M.A., Salazar, C., & Sumner, T. (2016). Fast and Easy Short Answer Grading with High Accuracy. HLT-NAACL.

[4] Mohler, Michael & Mihalcea, Rada. (2009). Text-to-Text Semantic Similarity for Automatic Short Answer Grading.. EACL 2009 - 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings. 567-575. 10.3115/1609067.1609130.

[5] Stanford CoreNLP. Povzeto iz: https://stanfordnlp.github.io/CoreNLP/

[6] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of the 21st national conference on Artificial intelligence - Volume 1 (AAAI'06), Anthony Cohn (Ed.), Vol. 1. AAAI Press 775-780.

[7] Žiga Simončič, Klemen Randl, ONJ - Seminar 2, (2019), Github repozitorij, https://github.com/venom1270/onj-seminar2