

Sistem za ocenjevanje esejev na podlagi koherence in semantične skladnosti

Automated Essay Evaluation System Based on Coherence and Semantic Consistency

Žiga Simončič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Večna pot 113, 1000 Ljubljana
zs3179@student.uni-lj.si

Zoran Bosnić

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Večna pot 113, 1000 Ljubljana
zoran.bosnic@fri.uni-lj.si

POVZETEK

V članku opisujemo implementacijo sistema za ocenjevanje esejev v angleškem jeziku. Zgledujemo se po metodologiji obstoječega sistema, ki poleg ocenjevanja sintakse uporablja tudi mere koherentnosti in semantične skladnosti. Metodologijo implementiramo v grafičnem okolju Orange, s prijaznim vmesnikom, opcijsko uporabo vektorskih vložitev za predstavitev besedila in možnostjo nadaljnega razvoja sistema. Sistem evalviramo na podatkih dostopnih na spletnem mestu Kaggle in, kolikor je mogoče, rezultate primerjamo z rezultati dosedanje metodologije in jih podrobno analiziramo. Poglobimo se tudi v izbiranje atributov za izboljšanje rezultatov. Glavni prispevki dela obsegajo (1) implementacijo sistema, (2) enostavnost uporabe in (3) izboljšave dosedanjega dela, vključno z dodatnimi računskimi opcijami in podrobno analizo izbiranja atributov za izboljšanje rezultatov.

KLJUČNE BESEDE

ocenjevanje esejev, semantična skladnost, Orange

ABSTRACT

In this paper we describe an implementation of an essay grading system. We lean heavily on the methodology of an existing system, which, besides using syntactical measurements, also uses coherence and semantic consistency measures. We implement the methodology in the Orange data mining tool, with a friendly user interface, optional use of word embeddings for word representation and the possibility for further developments of the system. The system is evaluated on public datasets from the Kaggle website. The results are to the most possible extent compared with the results of the existing methodology and analyzed in detail. We also compare several attribute selection methods, which improve our results. Main contributions of this work are comprised of (1) implementation of the system, (2) ease of use and (3) improvements upon previous work, including additional computing options and detailed attribute selection analysis.

KEYWORDS

automated essay evaluation, semantic consistency, Orange

1 UVOD

Učitelji v izobraževalnih ustanovah so odgovorni za predajanje znanj velikemu številu učencev. Del učnega procesa je tudi pisanje esejev, ki jih morajo učitelji prebrati in oceniti. Ocenjevanje esejev ni le časovno potratno, ampak potencialno tudi nekoliko pristransko. Naloga učitelja je tudi, da napake označi, popravi in komentira celotno delo.

S pomočjo računalnika lahko ocenjevanje esejev olajšamo. Dandanašnji sistemi za ocenjevanje esejev (tudi komercialni) se

osredotočajo predvsem na sintaksno analizo, premalo pozornosti pa posvečajo semantiki [6]. To slabost obstoječih sistemov rešuje sistem SAGE, ki ga Zupanc opisuje v svoji disertaciji [5]. SAGE dosega zavidljivo napovedno točnost v primerjavi z ostalimi sodobnimi sistemi, vendar je trenutna implementacija sistema v prototipni fazi in ni zrela za produkcijo.

Glavni cilj dela je bila implementacija sistema na način, da bo uporabnikom čimbolj dostopen, enostaven in prijazen za uporabo. Da zadostimo tem ciljem, smo se odločili za implementacijo v programskem okolju Orange,¹ ki je namenjen hitremu prototipiranju modelov in raziskovanju podatkov, namenjen tako začetnikom kot zahtevnejšim uporabnikom. Sistem je v Orange-u implementiran v obliki gradnikov (angl. widgets). Med seboj jih lahko povezujemo in kombiniramo, tako da smo uvoz datotek, gradnjo in testiranje modelov prepustili gradnikom, ki so v Orange-u že implementirani. Skupno smo implementirali tri gradnike – prvi implementira vse atributske funkcije, vključno s koherenco, drugi implementira sistem za analizo semantične skladnosti, tretji pa je namenjen evalvaciji modela po kvadratno uteženi kapi.

Sistem Zupanc [6] temelji na ekstrakciji različnih atributov iz podanih besedil (esejev) in se loči na tri (pod)sisteme: *AGE*, *AGE+* in *SAGE*. Oznaka "sistem Zupanc" predstavlja njeno implementacijo vseh teh treh sistemov. Vsak sistem nadgradi prejšnjega z dodatnimi atributi. Sistem *AGE* predstavlja skupek atributov osnovne sintaktične statistike, berljivostnih, leksikalnih, slovničnih in vsebinskih mer. To obsega različne značilnosti besedila, vse od osnovnih, kot so število znakov, besed itd., pa do števila slovničnih napak in računanje podobnosti z ostalimi esei. Skupno ta sistem zajema 72 različnih atributov, v prispevku tega članka pa smo temu sistemu dodali še pet novih atributov (št. znakov brez presledkov in štiri dodatne attribute, ki štejejo število posameznih oblikoskladenjskih oznak). Skupno torej 77 atributov.

Atributom sistema *AGE* dodamo attribute za merjenje koherence in s tem dobimo sistem *AGE+*. Koherenco merimo tako, da besedilo najprej razdelimo na prekrivajoče se odseke (drseče okno) in posamezne odseke pretvorimo v večdimenzionalni prostor. V tem prostor lahko posamezne odseke primerjamo in z različnimi merami ocenimo konsistentnost besedila in tok misli. Število atributov za merjenje koherence je 29.

Če vsem zgornjim atributom dodamo še nabor treh atributov, ki jih pridobimo s preverjanjem semantične skladnosti, govorimo o sistemu *SAGE*. Sistem za zaznavanje semantičnih napak v ozadju uporablja ontologijo, kateri postopoma dodajamo dejstva, ki jih izluščimo iz besedila. Z logičnim sklepanjem nato ugotovimo, če so trditve iz besedila logično konsistentne ali ne. To nam prinese tri dodatne attribute in možnost povratne informacije, v katerih povedih je prišlo do semantičnega neskladja.

¹<https://orange.biolab.si/>

2 SORODNA DELA

V sklopu svojega dela se je Zupanc [5] osredotočila na (v času njenega raziskovanja že zaključeno) tekmovanje avtomatskega ocenjevanje esejev, ki ga je gostil Kaggle.² Na tem tekmovanju so pomerili različni sistemi, s katerimi je Zupanc primerjala svoj sistem. Najboljša mesta na končni lestvici so večinoma zasedali komercialni sistemi za ocenjevanje esejev, nekaj pa je bilo tudi po meri narejenih uporabniških modelov. Komercialni sistemi kot so PEG,³ e-rater⁴ in IntelliMetric⁵ imajo že dolgo zgodovino in s tem velik tržni delež ter izpopolnjen finančni model. V času raziskovanja noben od naštetih ni ponujal brezplačne verzije sistema. Podrobno razčlenitev modelov in splošen opis njihovega delovanja najdemo v delih Zupanc [5] ter Zupanc in Bosnić [6].

V zadnjem času se na različnih področjih čedalje bolj uveljavljajo nevronske modeli, zato smo pogledali in testirali nekaj izvedb. Martinc in sod. [3] opisujejo uspešnost treh različnih nevronske modelov pri ocenjevanju besedil, ki sicer niso eseji. Tudi Taghipour in Tou Ng [4] sta primerjala različne nevronske modele za ocenjevanje esejev (na istih podatkih kot mi). Najboljši model dosega skoraj tak rezultat, kot mi. Alikaniotis in sod. so objavili članek [1], kjer so tudi testirali uspešnost različnih nevronske modelov na enaki podatkovni zbirki esejev, kot smo jo uporabljali mi.

3 OPIS IMPLEMENTACIJE IN METODE

3.1 Uporabljena orodja

Celoten sistem smo implementirali z uporabo orodja za podatkovno rudarjenje Orange v programskem jeziku Python. Glavne uporabljene knjižnice za razčlenitev besedila in izračun atributov so NLTK,⁶ SpaCy,⁷ scikit-learn⁸ in language-check⁹ za zaznavanje pravopisnih napak.

Za delo z ontologijami smo uporabili knjižnico rdflib¹⁰ in zunanja sistema (v smislu samostojna lokalna programa) ClausIE (na voljo tudi OpenIE5.0) in Hermit.¹¹

3.2 Implementacija gradnikov v Orange

Skupno smo razvili tri gradnike, ki zajemajo celoten opisan sistem. Slika 1 prikazuje vse tri gradnike, ki so opisani v nadaljevanju.

Prvi gradnik je namenjen izračunu vseh različnih mer. To so osnovne (plitke) statistične mere, mere berljivosti, leksikalne mere, slovnične mere, vsebinske mere in mere koherentnosti. Gradnik predstavlja sistema AGE in AGE+, odvisno od uporabniške izbire atributov, ki naj se izračunajo. Če označimo izračun vseh atributov, razen atributov za koherenco, govorimo o sistemu AGE, z dodanimi atributi za koherenco pa govorimo o sistemu AGE+. Ker je računanje nekaterih naprednih mer bolj zahtevno, se lahko uporabnik odloči za izračun kakršnekoli kombinacije naštetih šestih skupin mer. Za vsebinske mere in mere koherentnosti je na voljo dodatna izbira metode pretvorbe besedila v večdimenzionalni vektorski prostor. Tu podpiramo dve metodi: statistično pretvorbo TF-IDF in vektorske vložitve GloVe (v dveh izvedbah: SpaCy in Flair).

²<https://www.kaggle.com/>

³<https://www.measurementinc.com/products-services/automated-essay-scoring>

⁴<https://www.ets.org/>

⁵<http://www.intellimetric.com/direct/>

⁶<https://www.nltk.org/>

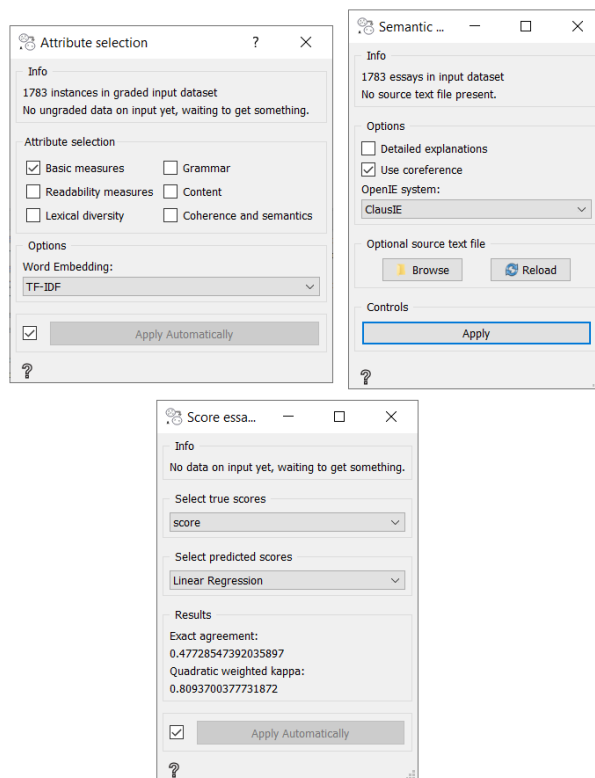
⁷<https://spacy.io/>

⁸<https://scikit-learn.org/stable/>

⁹<https://pypi.org/project/language-check/>

¹⁰<https://rdflib.readthedocs.io/en/stable/>

¹¹<http://www.hermit-reasoner.com/>



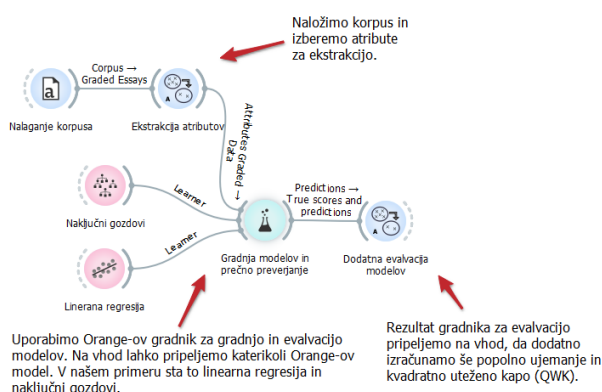
Slika 1: Prikaz vseh treh gradnikov

Gradnik ima tri vhode:

- (1) vhod za ocenjene eseje,
- (2) vhod za neocenjene eseje in
- (3) vhod za izvorno besedilo.

Vhoda za ocenjene in neocenjene eseje sta namenjena učni množici ocenjenih esejev in množici neocenjenih esejev, ki jim hočemo napovedati ocene. Na obeh množicah se izračunajo enaki atributi. Attribute ocenjenih esejev uporabimo za gradnjo modela. Vhod za izvorno besedilo je neobvezen in predstavlja izhodiščno zgodbo, knjigo ali dejstva, ki naj bi jih pisec esejja poznal. Če so eseji osnovani na podlagi nekega izvornega besedila, ga povežemo na ustrezen vhod in s tem izračunamo dodaten atribut (podobnost eseja z izvornim besedilom). Gradnik ima dva izhoda, in sicer izhod za izračunane attribute ocenjenih esejev in izhod za izračunane attribute neocenjenih esejev. To nam omogoča, da podatke ustrezno nastavimo kot vhode v ostale Orange-ove gradnike.

Drugi gradnik obsega delo in iskanje semantičnih neskladnosti z ontologijo. Predstavlja izračun dodatnih atributov, ki jih prinaša sistem SAGE. Gradnik je samostojen zaradi velike računske in časovne zahtevnosti. Ima dve nastavitvi: ali želimo uporabiti razreševalnik koreferenc in ali želimo, da se nam za semantične napake vrne podrobna razlaga. Uporaba koreferenc je priporočljiva, saj je v primerih posrednega navezovanja na različne pojme v besedilu to edini način zajetja celotne semantične informacije. Izberemo lahko tudi izvorno besedilo ali zgodbo, s katerim se razširi ontologijo, tako da ta vključuje tudi vsebino osnovnega besedila. To besedilo se bo obdelalo pred vsem ostalim, izluščene trojice pa bodo dodane v ontologijo. Razširjena ontologija se bo uporabila za preverjanje skladnosti esejev. Če



Slika 2: Primer uporabe sistema AGE/AGE+

izvornega besedila ne dodamo, se za preverjanje skladnosti normalno uporabi osnovna ontologija (ontologija COSMO). Gradnik ima samo en vhod — vhod za eseje ter en izhod — tabela treh atributov o številu posameznih napak in niz z osnovno razlago ter dodatni stolpec s podrobno razlago, če je ta izbrana.

Tretji gradnik je namenjen evalvaciji napovedanih ocen in pravih ocen esejev. Ker Orange ne podpira mer za izračun natančnega strinjanja (angl. *exact agreement*) in kvadratne utežene kape (angl. *quadratic weighted kappa - QWK*), smo naredili gradnik, ki prejme tabelo z napovedanimi ocenami in praviimi ocenami. Zgledovali smo se po izhodu gradnika *Test and Score* — za zagotavljanje interoperabilnosti lahko ta izhod vezemo neposredno na vhod našega gradnika, kjer se izračunata prej omenjeni meri.

Uporaba gradnika za izračun atributov in evalvacijo modela s kvadratno uteženo kapo je prikazana na Sliki 2.

3.3 Semantična analiza

Eden glavnih prispevkov dela Zupanc in Bosnić [6] je uporaba ontologij za ugotavljanje semantične skladnosti. Ta postopek je uporaben na dva načina: z njim pridobimo nekaj dodatnih atributov, ki jih lahko uporabimo pri napovedovanju ocen esejev, dodatno pa nam ta postopek tudi sporoči, kje se nahajajo semantične napake. Slednja funkcionalnost je zelo pomembna, saj tako učenec prejme neposredno informacijo o napakah v eseuju.

Postopek temelji na uporabi ontologije, v katero postopoma dodajamo v relacije strukturirane stavke in sproti preverjamo skladnost ontologije. Osnovna struktura ontologije je predstavljena s "trojicami" v obliki (*osebek, relacija, predmet*). Relacija lahko predstavlja omejitev, konceptualno povezavo (npr. (*Alice, isMotherOf, Bob*)) ali definira tip. V implementaciji smo za predstavitev trojic uporabili jezik RDF, ki je podoben jeziku OWL, vendar ni logični jezik. Uporabili smo ontologijo COSMO (angl. *Common Semantic Model*). Predstavljena je v semantičnem jeziku OWL¹² (*Web Ontology Language*), ki omogoča gradnjo kompleksnih shem različnih konceptov, dejstev in medsebojnih relacij. V primeru, da bi hoteli ontologiji dodati dodatna specifična znanja, to lahko storimo. V našem primeru je poleg nekaterih esejev tudi izvorno besedilo, na podlagi katerega so bili eseji spisani. Izvorno besedilo dodamo v ontologijo pred esejem in po enakem postopku kot eseji in je razložen spodaj.

Za posamezen esej poiščemo koreference v besedilu (angl. *co-reference resolution*). Ugotavljanje referenc nam omogoča odkrivanje posrednih referenc na določene entitete in zamenjavo z neposredno entiteto. Primer: "*Bob likes pizza. He eats it all the time.*" nadomestimo z "*Bob likes pizza. Bob eats pizza all the time.*"

Naslednji korak je razčlenitev besedila na posamezne povedi in ekstrakcija informacij s pomočjo sistema OpenIE (angl. *Open Information Extraction*). V tem koraku posamezne povedi pretvorimo v eno ali več trojic, ki opišejo relacije, izražene v povedi in so primerne za logično obdelavo. Za zgornji primer bi tako dobili dve trojici: (*Bob, like, pizza*) in (*Bob, eat, pizza*). Uporabili smo sistem za ekstrakcijo ClausIE [2], podpiramo pa tudi možnost uporabe sistema OpenIE5.¹³ Vse pridobljene trojice nato postopoma dodajamo v ontologijo, obenem pa preverjamo njeno skladnost. Za vsak element trojice poskušamo v ontologiji najti že obstoječ element. Pri tem preiščemo sopomenke, nadpomenke in protipomenke, v najslabšem primeru pa dodamo v ontologijo nov element. Po vsakem dodajanju elementov in trojic, preverimo skladnost ontologije. Skladnost preverjamo z logičnim sklepalnikom HermiT, ki vrača dva tipa napak. Prvi tip napak se zgodi, ko ima nek razred (*owl:Class*) prirerjene entitete, ki jih ne sme imeti (*unsatisfiable case*). Drugi tip napak pa se proži, ko se s sklepanjem ugotovi logična napaka — nekonsistentna ontologija (angl. *inconsistent ontology*). Do takšnih napak pride ponavadi zaradi neposrednih nasprotij (npr. *owl:disjointWith*) med dvema relacijama, ki pravi, da entiteta ne more imeti obeh relacij hkrati.

Na podlagi povzročenih tipov napak osnujemo tri dodatne attribute, ki jih lahko uporabimo pri napovedovanju ocen esejev: število neizpolnjenih primerov (pri dodajanju novih entitet v ontologijo), število napak nekonsistentne ontologije (pri dodajanju trojic) in vsota obeh prejšnjih.

3.4 Rezultati

Sistem smo testirali na podatkih že nekaj let starega tekmovanja ASAP na spletni strani Kaggle.¹⁴ Podatki obsegajo osem različnih podatkovnih zbirk (oz. devet, ker se druga zbirka ocenjuje po dveh kriterijih). Tema esejev v vsaki podatkovni zbirki je različna. Zbirke so razdeljene na učno, validacijsko in testno množico, vendar ocene validacijske in testne množice niso na voljo, zato smo za evalvacijo našega sistema uporabili 10-kratno prečno preverjanje. Razpon ocen je v vsaki zbirki različen, gibljejo se od 0–4, pa vse do 0–60. Za oceno modelov smo uporabili mero kvadratno utežene kape (angl. *quadratic weighted kappa*), ki upošteva razpon ocen in vrne relativno ujemanje napovedane ocene z dejansko oceno. Sistem smo testirali na modelu linearne regresije in naključnih gozdov. Bolje se je odrezala linearna regresija, zato smo se nanjo osredotočili v nadaljnjih eksperimentih. Uporabili smo regularizacijo L2 s parametrom $\alpha = 0,02$.

Na začetku smo modele gradili na celotnem naboru izračunanih atributov. Ker sistem AGE+, domnevno zaradi prevelikega števila atributov (106), ni dosegal boljših rezultatov od sistema AGE, smo preizkusili nekaj metod za izbiranje atributov. Glavni metodi naše analize sta bili vnaprejšnje izbiranje atributov (angl. *forward attribute selection*) in izločanje atributov (angl. *backward feature elimination*). Obe metodi sta izboljšali rezultat. Uporabili smo jih skupaj z 10-prečnim preverjanjem. Na vsaki iteraciji prečnega preverjanja smo dodali/odstranili posamezne attribute in glede na povprečje preko vseh iteracij dodali/odstranili atribut z največjim/najmanjšim prispevkom. To smo ponavljali, dokler ni bilo

¹²<https://www.w3.org/OWL/>

¹³<https://github.com/dair-iitd/OpenIE-standalone>

¹⁴<https://www.kaggle.com/c/asap-aes>

Tabela 1: Primerjava rezultatov brez izbiranja atributov naše implementacije sistemov AGE in AGE+ (TF-IDF), primerjava s sistemom Zupanc (AGE) in strnjeni rezultati izbiranja ter izločanja atributov na sistemu AGE+

	Brez izbiranja			Izbiranje	Izločanje
	AGE	AGE+	Zupanc (AGE)		
DS1	0,8358	0,8343	0,8447	0,8369	0,8439
DS2a	0,7001	0,7073	0,7389	0,7158	0,7324
DS2b	0,6789	0,6676	0,5386	0,6941	0,7028
DS3	0,6578	0,6622	0,6591	0,6656	0,6958
DS4	0,7536	0,7547	0,7174	0,7619	0,7769
DS5	0,7964	0,7955	0,7949	0,8028	0,8122
DS6	0,7734	0,7675	0,7636	0,7771	0,7871
DS7	0,8071	0,8034	0,7888	0,8083	0,8183
DS8	0,7479	0,7428	0,7738	0,7681	0,7717
AVG	0,7501	0,7484	0,7356	0,759	0,7712

več izboljšanja. Pri analizi smo opazili, da je nabor atributov, ki pride v končni izbor, relativno majhen. Ugotovili smo, da je zaradi prečnega preverjanja velika možnost, da s trenutnim naborom atributov pridemo v lokalni optimum. Zaradi povprečenja čez vse iteracije lahko nek atribut v prvi iteraciji izboljša rezultat, v drugi pa poslabša, in je v povprečju označen kot neprimeren. Za izogibanje tem lokalnim optimumom smo implementirali mejo, kolikokrat se lahko v povprečju rezultat poslabša, preden nabor atributov označimo kot končen. S tem smo kratkoročno poslabšali rezultat, vendar dolgoročno ustvarili kombinacijo atributov, ki dajejo v povprečju boljši rezultat. S to metodo izogibanja optimumov smo še dodatno izboljšali končne rezultate, ki so strnjeno prikazani v Tabeli 1. Pri izbiranju in izločanju atributov je AGE izpuščen, saj AGE+ v obeh primerih dosega boljše rezultate. Ker testni podatki niso več na voljo, smo naše rezultate s sistemom Zupanc lahko primerjali le s primerjavo sistemov AGE z 10-kratnim prečnim preverjanjem. Vidimo, da dosegamo zelo podobne rezultate, kot sistem Zupanc oz. jih nekoliko presehamo. Z ustreznim izbiranjem atributov pa naš rezultat še dodatno izboljšamo.

Sistem SAGE smo iz tabele izpustili, saj so rezultati z izločanjem atributom le malenkost boljši od sistema AGE+, prav tako pa smo ga uporabili le na podatkovnih zbirkah, ki so vsebovale izvorno besedilo (samo štiri zbirke). Kljub temu pa sistem SAGE ob zaznanem semantičnem neskladju nudi izpis povratne informacije. Primer v nadaljevanju prikazuje delovanje razreševalnika koreferenc in odkrivanje semantičnih napak. Zaradi korenjenja so nekatere besede v razlagi lahko odsekane. Vhod "George likes basketball and doesn't like sports.", sproži napako z razlago: "Relation 'George likes basketball and George doesn't like sports.' is inconsistent with a relation in ontology: 'George likes basketball and George doesn't like sports.'" in podrobno razlago: "Relation not consistent: Georg likes Basketball. Relations doesNotLike and likes are opposite/disjoint. Relation not consistent: Georg doesNotLike Basketball.". Osnovna razlaga deluje na ravni povedi in nam v tem primeru pove, da je poved v nasprotju sama s sabo. Podrobna razlaga pravi, da George ima in nima rad košarke. Beseda "sports" se v podrobni razlagi ne pojavi, ker je košarka podrazred športa in tam najprej pride do nasprotja.

Omenili bi še primerjavo našega sistema z omenjenimi nevronske modeli. Model Taghipour in Tou Ng [4] dosega podobne rezultate, kot naš sistem (nekaj pod 0,77). Alikaniotis in sod. [1] opisujejo, da njihov model dosega rezultat 0,96, vendar sumimo na nekaj nepravilnosti, ki izvirajo iz napačne uporabe mere za

ocenjevanje modelov (kvadratno utežene kape). Sumimo, da so za učenje svojega modela uporabili vse podatkovne zbirke skupaj, saj je njihov rezultat v območju skoraj 100% natančnosti (0,96), z dvakrat večjo absolutno napako (RMSE), kot naš model, ki ima rezultat približno 0,77. Z uporabo vseh zbirk na našem sistemu tudi dobimo tako visok rezultat (0,97 in 0,94, odvisno od modela).

4 ZAKLJUČEK

V sklopu tega dela smo implementirali sistem za ocenjevanje esejev po zgledu dela Zupanc [5] v programskem okolju Orange. Implementacija v okolju Orange omogoča enostavno uporabo sistema in združljivost z že implementiranimi funkcionalnostmi Orange-a. Sistemu smo dodali nekaj novih atributov in možnost predstavitve besed z vektorskimi vložitvami GloVe. Naša implementacija sistema je na voljo na repozitoriju git.¹⁵ Sistem temelji na ekstrakciji velikega števila atributov iz besedil in nato izboru najboljšega nabora za določeno podatkovno zbirko. Inovativni del preteklega dela, ki je vključen tudi v naši implementaciji, je dodaten sistem za preverjanje semantične skladnosti, s pomočjo katerega nabor atributov dodatno obogatimo, obenem pa imamo možnost, da nam sistem izpiše vse zaznane semantične napake oz. neskladja. Prispevek tega članka predstavlja tudi primerjava tehnik izbiranja atributov in primerjava rezultatov s preteklim delom. Sistem bi bilo smiselno preizkusiti tudi z drugimi napovednimi modeli, saj smo se v našem delu najbolj osredotočili le na linearno regresijo in naključne gozdove. Zanimiv izziv bi bil tudi prilagoditev sistema za slovenski jezik, ker je jezik sintaktično kompleksnejši, orodja za obdelavo besedil pa še niso tako zrela kot za angleški jezik.

ZAHVALA

Zahvaljujemo se sodelavcem Laboratorija za bioinformatiko na Fakulteti za računalništvo in informatiko za podporo in nasvete pri implementaciji sistema v programskem okolju Orange.

LITERATURA

- [1] Dimitrios Alikaniotis, Helen Yannakoudakis in Marek Rei. 2016. Automatic text scoring using neural networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. doi: 10.18653/v1/p16-1068. <http://dx.doi.org/10.18653/v1/P16-1068>.
- [2] Luciano Del Corro in Rainer Gemulla. 2013. ClausIE: Clause-Based Open Information Extraction. V *Proceedings of the 22nd international conference on World Wide Web*, 355–366.
- [3] Matej Martinc, Senja Pollak in Marko Robnik-Šikonja. 2019. Supervised and unsupervised neural approaches to text readability. *arXiv preprint arXiv:1907.11779*.
- [4] Kaveh Taghipour in Hwee Tou Ng. 2016. A neural approach to automated essay scoring. V *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, (november 2016), 1882–1891. doi: 10.18653/v1/D16-1193. <https://www.aclweb.org/anthology/D16-1193>.
- [5] Kaja Zupanc. 2018. Semantics-based automated essay evaluation. Doktorska disertacija. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani.
- [6] Kaja Zupanc in Zoran Bosnić. 2017. Automated essay evaluation with semantic analysis. *Knowledge-Based Systems*, 120, 118–132.

¹⁵<https://github.com/venom1270/essay-grading>