SloTex odprtokodno ogrodje za procesiranje slovenskega naravnega jezika

Klara Eva Kukovičič[1], Sonja Debevec[1], Mark Juvan[2], Jakob Bernik[2], Aljaž Trebušak[3], Peter Čebašek[3], Simon Dobrišek[3], Tadej Justin[4]

[1]*Filozofska fakulteta, Aškerčeva cesta 2, 1000 Ljubljana*
[2]*Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana*
[3]*Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana*
[4]*Medius d.o.o, Tehnološki park 21, 1000 Ljubljana*
E-pošta: tadej.justin@medius.si

SloTex - Open Source Framework for Slovenian Natural Language Processing

High-tech companies nowadays use language technologies every day. Algorithms are embedded in applications that enable the processing of large amounts of data. There are many open source frameworks on the market that enable the use of language technologies for many languages. Since Slovene has a relatively small number of speakers and consequently lower market relevance, it is often neglected. In Slotex project we developed an open-source language-technology web framework that enables natural language processing for use in enterprise environments. The main goal of the project is not only to develop a reliable web-based system that implements open source frameworks such as OpenNLP and CoreNLP, but also to develop initial language models for Slovenian language. In this paper we present evaluation of the Slovenian Name Entity Recognition tagging for only person tags and compare it to our own implemented solution based on Levenshtein distance and external lexicons. Proposed solution more than doubles true positive entities, but also significantly increases false negatives.

1 Uvod

Procesiranje naravnega jezika se dandanes uporablja v že skoraj vsaki spletni aplikaciji, platformi ali v spletnih iskalnikih. Tako skoraj vsak uporabnik spleta nevede uporablja vsaj nekaj algoritmov za procesiranje naravnega jezika. Obsežna zgodovina področja procesiranja naravnega jezika (ang. Natural Language Processing, NLP) je narekovala razvoj različnih programskih orodij in knjižnic, ki omogočajo uporabo najbolj razširjenih operacij in algoritmov. Med najbolj znanimi so Natural Language Toolkit, NLTK [1], TextBlob [2], CoreNLP [3] in OpenNLP [4]. Funkcionalnosti, ki jih v določenem obsegu omogočajo tovrstne knjižnice, so tokenizacija, lematizacija, korenjenje, oblikoskladenjsko označevanje, klasifikacija, izdelava stavčnih dreves, izdelava n-gramov, dostop do korpusov besedil, funkcije za iskanje vzorcev v besedilu, štetje frekvenc besed. Večina navedenih funkcionalnosti je odvisnih od jezika. Če želimo uporabiti orodja in algoritme tudi za slovensko besedilo, je potrebno v večini primerov najprej pridobiti dobro označen korpus slovenskega besedila. Na podlagi označb lahko zgradimo slovensko podporo ali jezikovne modele, ki jih lahko kasneje uporabimo pri obdelavi poljubnega slovenskega besedila. Programske knjižnice se v raziskovalni skupnosti uporabljajo kot programska ogrodja, ki jih lahko relativno enostavno prilagodimo za specifičen jezik ob predpostavki, da razpolagamo z označenim korpusom. Nobena od zgoraj navedenih programskih knjižnic žal nima neposredno vgrajene podpore za slovenski jezik, zato moramo vsako pred uporabo v slovenščini prilagoditi in dodatno naučiti jezikovne modele za specifično funkcionalnost.

Klub temu vseeno obstaja kar nekaj knjižnic, ki so bile razvite v okviru različnih raziskovalnih projektov in so prilagojene izključno za delo v slovenskem jeziku. Ena takih je sistem Obeliks, ki omogoča lematizacijo in označevanje oblikoskladenjskih oznak za slovenska besedila [5]. Nekaj programske opreme s podporo za slovenski jezik pa je ponudil tudi projekt Sporazumevanje v slovenskem jeziku (SSJ).

V tem prispevku opisujemo naša prizadevanja pri razvoju odprtokodnega ogrodja za procesiranje naravnega jezika, ki je primarno namenjeno za uporabo v poslovnih okoljih in omogoča uporabniku razvoj lastnih jezikovnih modelov za slovenski jezik. Predstavimo pa tudi razvoj algoritma za označevanje oseb v slovenskem besedilu in uspešnost primerjamo z implementiranim sistemom v ogrodju OpenNLP [4]. Predstavljeni algoritem temelji na podlagi najmanjše razdalje med zunanjim leksikonom imen in Levenshteinovo razdaljo. S tem principom lahko omogočimo tudi v neoznačenem besedilu iskanje imenskih oznak oseb, kar je še posebej uporabno, če jezik, v katerem gradimo bolj napredne modele za označevanje imenskih entitet, nima obsežnega, ročno označenega korpusa.

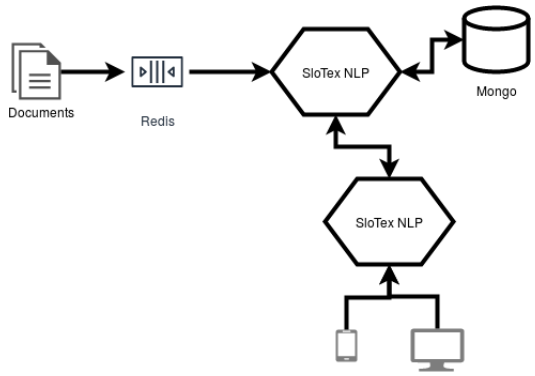
1.1 SloTex NLP

Trenutno je na trgu že več priznanih odprtokodnih ogrodij, ki omogočajo uporabo jezikovnih tehnologij v enem od svetovnih jezikov. Slovenščina je zaradi relativno majhnega števila govorcev in posledično manjše tržne relevantnosti pogosto zapostavljena. Z realizacijo aplikacije in vključitvijo priznanih ogrodij za procesiranje naravnega jezika pa smo tudi za slovenski jezik realizirali jezikovne modele, ki omogočajo njihovo neposredno uporabo za razčlenitev besedil, označevanje besednih vrst in označevanje imenskih entitet. Čeprav pridobljeni rezultati ne

presegajo najbolj naprednih sistemov, ki omogočajo procesiranje tudi slovenskega naravnega jezika, je ključen prispevek pri projektu aplikacija, ki jo lahko zlahka uporabimo tudi v poslovnih okoljih in omogoča hitro in enostavno obdelavo podatkov.

V okviru projekta "Po kreativni poti do znanja" (PKP), smo sestavili interdisciplinarno skupino šestih študentov/-k, dva študenta Fakultete za elektrotehniko v Ljubljani, dva študenta Fakultete za računalništvo in informatiko v Ljubljani in dve študentki Filozofske fakultete v Ljubljani. Ideja projekta je bila primarno udeležene študente in kasneje tudi širšo javnost navdušiti nad uporabo in razvojem aplikacij, povezanih z jezikovnimi tehnologijami. Projekt smo naslovili "Razvoj slovenskih jezikovno tehnoloških rešitev za uporabo v poslovnih informacijskih sistemih" z akronimom SloTex.

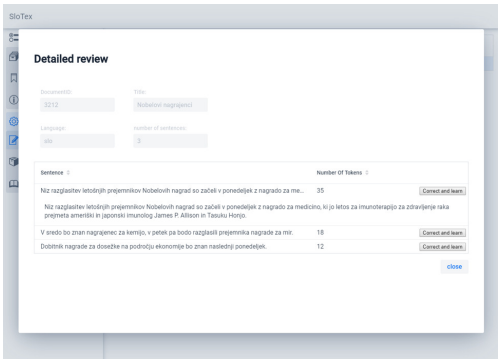
Cilj projekta je realizirati odprtokodno aplikacijo, ki ima prilagojene funkcije za procesiranje naravnega jezika za slovenščino in je uporabna tudi v poslovnih aplikacijah. V okviru projekta smo zasnovali aplikacijo na podlagi mikrostoritvene arhitekture, ki omogoča enostavno procesiranje podatkov, njihov pregled, učenje novih modelov in njihovo vrednotenje kar preko spletnega brskalnika. Slika 1 prikazuje osnovno zgradbo mikrostoritvene arhitekture.



Slika 1: Mikrostoritvena arhitektura sistema SloTex NLP.

Aplikacijo smo razvili kot prosto dostopno orodje in jo objavili na spletu v treh projektih: "SloTex NLP core"[1], "SloTex NLP web"[2] in "SloTex NLP entity"[3]. Procesiranje besedila se izvaja neposredno v zaledni komponenti "SloTex NLP core", ki omogoča procesiranje obsežnejših besedil, ali pa samo stavkov, preko spletnih servisov na podlagi REST priporočil in vmesnega pomnilniškega strežnika Redis. "SloTex NLP entity" pa je komponenta, ki vsebuje ločeno hrambo podatkovnih modelov, ki jih uporablja aplikacija. Spletni uporabniški vmesnik je zajet v projektu SloTex NLP web", ki združuje različna jezikovna orodja za obdelavo slovenskega jezika.

Uporabniški vmesnik sestoji iz dveh delov, in sicer Administracije (Administration) in Učenja (Train). Administracijski del uporabniku omogoča pregled že obdelanih besedilnih dokumentov, označenih entitet in njihove podrobnejše podatke ter podatke o tem, koliko dokumentov je še v čakalni vrsti ter koliko jih je namenjenih za obdelavo. Poslovni uporabniki pričakujejo, da lahko v sistem vnesejo celotne dokumente, kot izhod pa pridobijo označene entite v besedilu z vnaprej podanim šifrantom. Tako smo tudi zasnovali sistem, ki omogoča vnos celotnih dokumentov v obliki (.doc ali .txt), ki jih uporabnik lahko s pomočjo enostavnega programa ali spletnega vmesnika naloži v sistem. Glede na vnaprej določen šifrant zaledna aplikacija označi besedilo in ponudi označeno besedilo uporabiku za prevzem. Uporabnik ima možnost tudi pregledati dodatno analizo, ki nudi pregled pojavnic s pripadajočo besedno vrsto, lemo in označeno entiteto. Omogočeno je tudi ročno urejanje vseh avtomatsko določenih kategorij. Če si uporabnik želi, jih lahko pregleda in ročno uredi oz. popravi. Tako popravljen oz. označen dokument lahko tudi vključi v učno množico in ponovno nauči jezikovni model. S tem lahko sam pripomore k izboljšanju natančnosti avtomatsko označenih kategorij. Sistem tako omogoča gradnjo modelov na obstoječih zbirkah, obenem pa nudi uporabniku izboljšave modelov tudi z lastnimi označenimi dokumenti. Slika 2 prikazuje uporabniški vmesnik za urejanje lastnega besedila.



Slika 2: Prikaz urejanje in označevanja dokumentov.

2 Metodologija označevanja imenskih entite

Za doseganje dobrih rezultatov je ključnega pomena obsežna podatkovna baza. Po pregledu že obstoječih orodij za slovenščino, med katerimi so najvidnejši RELDI-tagger [6] in Obeliks, smo ugotovili, da se za vrednotenje sistemov uporabljajo predvsem besedilni korpusi, ki so bili razviti za potrebe projekta Jezikoslovno označevanje slovenskega jezika (JOS) in Sporazumevanje v slovenskem jeziku (SS). Kot šolski besedilni korpus slovenskega jezika se v zadnjem času uporablja predvsem SSJ500k in Sloleks. Ker smo želeli pridobiti čim bolj primerljive podatke, smo se odločili, da za vrednotenje razvitega algoritma označevanja osebnih imenskh entite uporabimo korpus SSJ500k. To je ročno pregledan korpus s približno pol milijona besed in je dostopen pod licenco Creative Commons za učenje modelov za označevanje entitet.

Prepoznavanje imenskih entitet vključuje proces iden-

[1] https://github.com/MediusInc/slotex-nlp-core
[2] https://github.com/MediusInc/slotex-nlp-web
[3] https://github.com/MediusInc/slotex-nlp-entity

tifikacije imenske entitete iz besedila in klasifikacijo teh entitet v vnaprej določene razrede razvrščanja [7]. Besedilni korpusi za ta namen ponavadi predvidevajo štiri osnovne razrede razvrščanja: osebe, lokacija, organizacija in nedoločeno. Ta prispevek se osredotoča le na razpoznavanje kategorije oseb v slovenskem jeziku. Za proces identifikacije imenskih entitet se v najsodobnejših sistemih uporabljajo predvsem postopki strojnega učenja, kot so, na primer, skriti Markovi modeli, pogojna naključna polja in principi maksimalne entropije (ang. maximum entropy) [8]. Naštete metode temeljijo na nadzorovanem učenju, za kar je potreben korpus besedil. Drugi sistemi pa za prepoznavanje imenskih entitet uporabljajo tudi eksplicitno predznanje, ki temelji na leksikonih imen. Ker pa tovrstni sistemi ne zaznajo entitete, če se ta ne nahaja v leksikonih, se jih v praksi pogosto kombinira s sistemom, ki je osnovan na strojnem učenju [9]. V prispevku se nismo posvetili optimizaciji obstoječih algoritmov ali specifičnih prilagoditev za slovenski jezik, pač pa smo si zastavili vprašanje, na kakšen način lahko ponudimo obstoječim algoritmom avtomatsko označena besedila. Ta besedila pa so kasneje lahko uporabljena v učni množici pri nadzorovanem učenju.

Pristop temelji na uporabi leksikonov in uporabi klasične metrične razdalje, ki se uporablja za primerjavo med nizi Levenshteinove razdalje [10]. Razdaljo med dvema nizoma tako določimo s številom minimalnih operacij, ki jih potrebujemo za preoblikovanje enega niza v drugega, pri čemer so operacije definirane kot vrivanje, brisanje in zamenjave znakov, operacije so na primeru prikazane v tabeli 1. Večja vrednost razdalje ponazarja večje razlike, manjša pa večjo podobnost med primerjalnima nizoma.

Tabela 1: Primer možnih operacij za izračun Levenshteinove razdalje.

Operacija	Niz 1	Niz 2
vrivanje	Marko	Markov
brisanje	Marko	Mark
zamenjava	Marko	Marka

Zasnovali smo algoritem, ki najprej normalizira in tokenizira besedilo. Vsako pojavnico nato primerja z imeni s seznama imen, ki ga določimo kot parameter algoritma. Za potrebe tega prispevka smo seznama imen in priimkov pridobili iz registra imen Statističnega urada Republike Slovenije. Seznam vsebuje 4175 moških in 4268 ženskih imen in 30667 priimkov.

Algoritem na podlagi Levenshteinove razdalje za vsako besedo iz besedila določi razdaljo med pojavnico in imeni s seznama. Če je razdalja med pojavnico in imenom s seznama manjša ali enaka 1, predpostavimo, da imata besedi isto lemo, posledično pa, da je pojavnica tudi imenska entiteta, saj leksikon vsebuje le spisek imenskih entitet. Pri preizkusih smo ugotovili, da lahko pri izračunu Levenshteinove razdalje utežimo začetek vsake pojavnice, saj se pri pregibanju besed v slovenščini spreminjajo le končnice. Le v nekaj redkih primerih pa tudi koreni besed. Če je algoritem torej naletel na pojavnico "Marka", je bila razdalja do imena 'Marko" zaradi uteži manjša od 1, saj je bila obtežena osnova imena, torej "Mark". Posledično je algoritem pojavnico prepoznal kot imensko entiteto s seznama.

Privzet algoritem za računanje Levenshteinove razdalje dovoljuje menjavo največ dveh zadnjih črk. V slovenskem jeziku pa je pri pregibanju možna menjava tudi večjega števila črk. Zato smo implementirali izboljšavo, ki temelji na upoštevanju pogostih končnic pri pregibanju samostalnikov in svojilnih pridevnikov. Ustvarili smo seznam vseh možnih končnic, ki se pojavljajo v slovenskem jeziku pri pregibanju samostalnikov ženskega, moškega in srednjega spola ter pri pregibanju svojilnih pridevnikov. Seznam končnic smo pridobili iz označenega oblikoslovnega slovarja Sloleks, ki je označen z oznakami, napisanimi v okviru projekta MULTEXT-East, pri čemer se oznake samostalnikov začenjajo s črko S, pridevniki pa s črko P. Lastno ime "Marko" bi bilo torej v Sloleksu označeno s Slmei, kar pomeni, da je Marko samostalnik (s), lastno ime (l), moškega spola (m), v ednini (e) in se v dotičnem primeru pojavi v imenovalniku (i). V slovarju so s tovrstnimi oblikoskladenjskimi oznakami označene vse pojavnice, zato smo napisali preprost program, ki nam je iz besedilnih podatkov izluščil vsa lastna imena in svojilne pridevnike. Iz izluščenih podatkov smo nato pridobili vse možne končnice in jih uporabili pri določanju imenskih entitet z Levenshteinovo razdaljo. Uporabljene končnice so prikazane v tabeli 2.

Tabela 2: Izluščene končnice iz korpusa Slolex za samostalnike in svojilne pridevnike.

Samostalniki	Svojilni pridevniki
-a -e -em -ema -es -ev -ga -h -i -ih -ja -je -jem -jema -jev -ji -jih -jo -ju -m -ma -mi -mu -om -oma -ov -ta -te -tem -tema -tev -ti -tih -tom -toma -tov -tu -u -v	-a -e -ega -em -emu -i -ih -im -ima -imi -o

Končnice smo uporabili za pravilno označevanje pregibnih oblik imen in priimkov, pri katerih pride do menjav treh ali štirih črk končnic na koncu niza. Če ima pojavnica veljavno končnico za ime ali priimek v tabeli 2, se vrednost končne Levenshteinove razdalje zmanjša za trimestno končnico 1,8, za štirimestne končnice pa 3,8. S tem smo omogočili menjavo treh in celo štirih črk v primerjalnih besedah. To dovoljuje detekcijo pregibnih oblik, pri katerih pride do menjave večjega števila črk, saj privzeti algoritem za računanje Levenshteinove razdalje dovoljuje menjavo največ dveh zadnjih črk.

3 Rezultati

Predlagan označevalnik imenskih entitet z oznako "osebašmo vrednotili na korpusu slovenskega besedila SSJ500k

z 10-kratnim navzkrižnim učenjem. Na vnaprej naključno izbranih desetih delih besedilnega korpusa SSJ500k, namenjenih za učenje in vrednotenje, smo izdelali in vrednotili označevalnik imenskih entitet v programskem ogrodju OpenNLP (TokenNameFinder). Z ogrodjem smo tako zgradili 10 modelov za označevanje imenskih entitet in jih vrednotili na pripadajočih testnih besedilih. Testne podatke (celotno zbirko SSJ500k) smo uporabili tudi za označevanje imenskih entitet s predlaganim algoritmom na podlagi Levenshteinove razdalje in leksikona imen in priimkov.

Rezultate poročamo z metriko za vrednotenje: natančnost, priklic in F1, ki so pogosto predpisane metrike ob različnih tekmovanjih [11].

Tabela 3: Rezulati vrednotenja sistema za označevanje imenskih entite

Sistem	**Natančnost**	**Priklic**	F_1
OpenNLP	$0,466$	$0,169$	$0,248$
Predlagan sistem	$0,173$	$0,457$	$0,251$

Tabela 3 prikazuje primerljive rezultate pri označevanju imenskih entitet oseb v besedilu. S predlaganim označevalnikom smo močno povečali število napačno označenih oseb, obenem pa smo povečali tudi število pravilno označenih oseb v primerjavi s privzetim označevalnikom za razpoznavanje besednih entitet v ogrodju OpenNLP, ki označi precej manj oseb pravilno, vendar ima tudi precej manjše število napačno označenih oseb v besedilu.

4 Zaključek

V prispevku smo predstavili odprtokodno aplikacijo za procesiranje naravnega jezika, ki ima tudi integrirano slovensko jezikovno podporo in uporabnikom ponuja možnosti dodatnega učenja jezikovnih modelov in dostop do sodobnih funkcionalnosti ogrodja OpenNLP. V prihodnosti se nadejamo še izboljšati uporabnost razvite aplikacije ter dodati še kakšno obstoječe ogrodje za procesiranje naravnega jezika z vključeno slovensko jezikovno podporo.

Poleg spletne aplikacije za enostavno uporabo jezikovnih tehnologij s slovensko jezikovno podporo, namenjene za uporabo v poslovnih okoljih, smo se posvetili rezultatom udejanjenega algoritma za označevanje oseb s pomočjo Levenshteinove razdalje ter leksikona imen in priimkov, pridobljenem na Statističnem uradu Republike Slovenije. Vrednotenje algoritma smo primerjali z izdelavo osnovnega označevalnika imenskih entitet z ogrodjem OpenNLP in pridobili primerljive rezultate. Ugotavljamo, da smo pri označevanju oseb na korpusu slovenskega besedila SSJ500k s predlaganim algoritmom označili 2,7-krat več pravilno označenih oseb, obenem pa smo tudi povečali število napačno označenih besed kar za faktor 11,2-krat.

Predstavljen algoritem pa v primerjavi z algoritmom OpenNLP za označevanje imenskih entitet krasi dejstvo, da za označevanje oseb v slovenskem besedilu ne potrebujemo obsežnega ročno označenega besedilnega korpusa, pač pa le prosto dostopen leksikon imen. Rezultati so pokazali, da se algoritem lahko primerja tudi z bolj naprednimi algoritmi, osnovanimi na statističnem modeliranju, kot to velja za ogrodje OpenNLP. Pomanjkljivost, ki se jo zaznavamo, pa je, da se predlagan algoritem ne zaveda besedilnega konteksta, zato je še vedno močno podvržen označevanju napačnih besed.

Če nam bo v prihodnosti dodatno uspelo izboljšati označevanje napačno označenih oseb, bo algoritem nedvomno uporaben za avtomatsko označevanje oseb v besedilu. V kombinaciji z naprednejšimi algoritmi osnovanimi na nadzorovanem stojnem učenju za razpoznavanje imenskih entitet pa lahko dodatno pripomore k izboljšanju natančnosti.

Zahvala

Ta prispevek je rezultat interdisciplinarne skupine projekta SloTex, katerega financiranje je podprl razpis Po kreativni poti do znanja 2016-2020 [4].

Literatura

[1] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[2] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey, *et al.*, "Textblob: simplified text processing," *Secondary TextBlob: Simplified Text Processing*, 2014.

[3] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.

[4] J. Baldridge, "The opennlp project," *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012)*, p. 1, 2005.

[5] M. Di Batista, *Označevanje imenskih entitet v pravnih besedilih*. PhD thesis, Univerza v Ljubljani, 2013.

[6] N. Ljubešić and T. Erjavec, "Corpus vs. lexicon supervision in morphosyntactic tagging: the case of slovene," in *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*, pp. 1527–1531, 2016.

[7] J. Flisar and M. Pavlinek, "Wikipedija kot vir znanja za iskanje imenskih entitet," *Elektrotehniski Vestnik*, vol. 84, no. 3, p. 108, 2017.

[8] D. Fišer, T. Erjavec, A. Z. Vitez, and N. Ljubešid, "Janes se predstavi: metode, orodja in viri za nestandardno pisno spletno slovenščino," *V: Jezikovne tehnologije: zbornik*, vol. 17, pp. 56–61, 2014.

[9] T. Štajner, T. Erjavec, and S. Krek, "Razpoznavanje imenskih entitet v slovenskem besedilu," *Slovenščina 2.0: empirical, applied and interdisciplinary research*, vol. 1, no. 2, pp. 58–81, 2013.

[10] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, pp. 707–710, 1966.

[11] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[4] http://www.sklad-kadri.si/si/razvoj-kadrov/po-kreativni-poti-do-znanja-pkp/