Orodja za procesiranje nestandardne slovenščine

Nikola Ljubešić, Tomaž Erjavec, Darja Fišer

Izvleček

Poglavje je posvečeno težavam, povezanim z avtomatskim procesiranjem nestandardnega jezika, in orodjem, ki smo jih razvili za reševanje teh težav. V poglavju obravnavamo merjenje standardnosti besedil, stavčno segmentacijo, normalizacijo, rediakritizacijo, oblikoskladenjsko označevanje in razpoznavanje imenskih entitet. Pokažemo, da se število napak, ki jih povzročijo orodja, naučena na standardnem jeziku, pri uporabi za nestandardni jezik sicer močno poveča, vendar predhodna normalizacija nestandardnih besedil ali prilagoditev orodij zanje bistveno povečata kvaliteto procesiranja nestandardnega jezika. Za ta namen potrebujemo dovolj ročno označenih nestandardnih besedil, ki jih nato uporabimo za učenje ali posodabljanje modelov za nadzorovano strojno učenje.

Ključne besede: jezikovne tehnologije, nestandardni jezik, normalizacija besedil, oblikoskladenjsko označevanje, razpoznavanje imenskih entitet

1 UVOD

Z vidika procesiranja podatkov sodi jezik med zahtevnejše naloge tudi zaradi znatne večpomenskosti. Ta naloga postane še težja, ko besedila odstopajo od pravopisnih in slovničnih norm, kar je pri jeziku na spletu zelo pogost pojav. V slovenščini najpogostejša odstopanja od jezikovne norme predstavljajo opuščanje strešic, nestandardno črkovanje in pogosta uporaba pogovornih izrazov (Fišer et al. 2015).

Tovrstni pojavi močno vplivajo na avtomatsko procesiranje besedil. Gimpel et al. (2011) poročajo, da so pri učenju in testiranju modela na podatkih iz korpusa Wall Street Journal pri pripisovanju oblikoskladenjskih oznak dosegli 97-% točnost, ko so isti model uporabili na besedilih s Twitterja, pa je točnost znašala le 85 %, kar predstavlja petkratno povečanje števila napak. Ljubešić et al. (2017) so izvedli eksperiment z oblikoskladenjskim označevanjem slovenščine, ki ima v primerjavi z angleščino veliko kompleksnejšo označevalsko shemo. Standardni označevalnik je pri testni množici s standardnimi besedili dosegel 94-% točnost, pri nestandardnih besedilih pa le 69-%, kar ponovno predstavlja petkratno povečanje števila napak.

V pričujočem poglavju predstavimo različne pristope za zmanjševanje težav pri procesiranju nestandardnih besedil. V naslednjem razdelku opišemo postopek identifikacije nestandardnih besedil, v tretjem razdelku pa prilagoditve orodja za segmentacijo, tj. ločevanje niza znakov na stavke in pojavnice. V naslednjih štirih razdelkih predstavimo dva pristopa, ki se pogosto uporabljata za nadaljnje procesiranje nestandardnih besedil (Eisenstein, 2013): (1) normalizacija besedil v standardno obliko in uporaba standardnih orodij ter (2) prilagoditev orodij za nestandardne vhodne podatke. Za prvi pristop opišemo postopek normalizacije, ki temelji na statističnem strojnem prevajanju na nivoju znakov, in orodje za rediakritizacijo besedil, za drugi pristop pa prilagoditve orodij za oblikoskladenjsko označevanje in razpoznavanje imenskih entitet. Čeprav se v poglavju posvečamo predvsem slovenščini, smo hkrati razvili tudi orodja za hrvaščino in srbščino, zato navajamo rezultate za vse tri jezike.

Večina orodij, ki jih predstavimo v poglavju, temelji na paradigmi nadzorovanega strojnega učenja, kar pomeni, da moramo podatke za obravnavani problem najprej ročno označiti. Če želimo denimo napovedati standardnost besedila, označimo vzorec besedil glede na njihovo standardnost. Drugi primer je pripisovanje oblikoskladenjskih oznak besedam v vzorcu. Pripisane oznake (npr. stopnja standardnosti ali oblikoskladenjske oznake) imenujemo *odvisne spremenljivke*, specifične spremenljivke, ki jih izluščijo razvita orodja, pa *neodvisne spremenljivke* ali značilke. Orodja nato modelirajo odvisnost med odvisnimi in neodvisnimi spremenljivkami. Ko je proces modeliranja, imenovan tudi *učni proces*, zaključen,

lahko orodja *napovejo* vrednosti odvisnih spremenljivk za nova besedila, in sicer tako, da iz teh besedil izluščijo neodvisne spremenljivke in uporabijo izdelani napovedni model.

2 NAPOVEDOVANJE STANDARDNOSTI

Splošna domneva je, da jezik na spletu odstopa od norme (Crystal 2011), vseeno pa ta pojav navadno ni kvantitativno izmerjen. Da bi to izboljšali, smo razvili posebno orodje – kolikor nam je znano, prvo te vrste –, ki napove stopnjo standardnosti določenega besedila (Ljubešić et al. 2015). Orodje je uporabno z dveh vidikov: (1) izboljša lahko postopek procesiranja korpusov, saj pomaga pri odločitvi, ali za izbrano besedilo uporabimo model za standardni ali nestandardni jezik, in (2) omogoča, da podatek o standardnosti besedil uporabimo v korpusnojezikoslovnih analizah. Drugi vidik je bil tudi naš glavni cilj pri gradnji orodja, torej omogočanje korpusnim jezikoslovcem, da se pri raziskovanju osredotočijo bodisi na standardni bodisi na nestandardni spletni jezik.

2.1 Določanje standardnosti besedila

Pojem standardnosti besedila razumemo kot avtorjevo upoštevanje jezikovnih norm, opisanih v pravopisnih, slovničnih in slogovnih priročnikih. Avtomatsko določanje standardnosti besedila ni lahka naloga. Medtem ko lahko večino pojavov obravnavamo kot eno dimenzijo besedila (kot na primer pri označevanju sentimenta), se izkaže, da standardnost zajema zelo raznolike značilnosti. Nekateri avtorji na primer uporabljajo standardno črkovanje, a opuščajo veliko začetnico. Drugi napravijo veliko tipkarskih napak, tretji pa se držijo standardne uporabe ločil, a uporabljajo pogovorno ali narečno besedišče ter oblikoskladnjo.

Da bi zagotovili pravšnje razmerje med ustreznostjo in kompleksnostjo oznak, smo se odločili za uporabo dveh dimenzij standardnosti: tehnično in jezikovno. Stopnja tehnične standardnosti besedila (okrajšana s »T«) pokriva uporabo velike začetnice, uporabo ločil, prisotnost tipkarskih napak ali ponovljenih znakov v besedi (npr. *napreeej*). Stopnja jezikovne standardnosti (okrajšana z »L«) upošteva črkovanje, besedišče, oblikoslovne lastnosti in besedni red. Za obe dimenziji smo uporabili tri vrednosti: 1 (povsem standardno), 2 (nekoliko nestandardno) in 3 (zelo nestandardno).

Sistem dveh dimenzij s tremi razredi je zasnovan tako, da označevalcem omogoča enostavno pripisovanje stopnje standardnosti, je dovolj informativen za orodja za procesiranje naravnega jezika za izbiro različnih metod normalizacije in hkrati

dovolj relevanten za jezikoslovce, da lahko filtrirajo besedila, ko raziskujejo nestandardni jezik. Primera z ekstremnimi vrednostmi na obeh dimenzijah sta podana v Tabeli 1.

Tabela 1: Stopnja standardnosti dveh besedil.

T1L3	Tehnično povsem standardno, jezikovno zelo nestandardno
Izvirnik	*Ma men se zdi tole s poimenovanji oz s poslovenjenjem imen mest čist mem.*
Standardizirano	*Meni se zdi to s poimenovanji oz. s poslovenjenjem imen mest čisto mimo.*

T3L1	Tehnično zelo nestandardno, jezikovno povsem standardno
Izvirnik	*se pravi,da predvidevaš razveljavitev*
Standardizirano	*Se pravi, da predvidevaš razveljavitev?*

2.2 Izdelava podatkovne množice

Podatke za eksperimente smo zajeli iz vmesne različice korpusa Janes, vsebujejo pa tri vrste besedil: tvite, objave na forumih in komentarje pod spletnimi novicami. Podatkovna množica za eksperimente vsebuje posamezna besedila, ki smo jih vzorčili iz korpusa po postopku, opisanem v nadaljevanju. Besedila so bila nato ročno označena.

Da bi zagotovili uravnoteženo množico podatkov, smo zbrali enak delež (eno tretjino) besedil za vsako vrsto, prav tako pa smo pri objavah na forumih in komentarjih pod novicami vključili enak delež besedil iz vseh šestih virov. Da bi iz korpusa, v katerem prevladuje standardni jezik, pridobili uravnoteženo podatkovno množico glede jezikovne (ne)standardnosti, smo na podlagi postopka normalizacije (Ljubešić et al. 2014) približno ocenili stopnjo (ne)standardnosti besedil. Za vsako besedilo smo izračunali razmerje med številom pojavnic, ki so bile v procesu avtomatske normalizacije spremenjene, in skupno dolžino besed. Besedila z razmerjem 0,1 ali manj smo obravnavali kot standardna, ostala pa kot nestandardna. To merilo smo določili ročno na podlagi pregledovanja podatkov. Podatkovno množico smo nato izdelali tako, da je vsebovala enako število domnevno standardnih in nestandardnih besedil. Poudariti je treba, da smo pri vzorčenju uporabili precej grobo metodo za zagotavljanje uravnotežene podatkovne množice in na ta način zaobšli razmeroma nizek odstotek nestandardnih besedil v celotnem korpusu. Za vzorčenje bi lahko uporabili tudi druga groba merila standardnosti, npr. delež besed zunaj besedišča (angl. *out-of vocabulaty ratio*) glede na leksikon standardnih besednih oblik.

2.3 Ročno označevanje in dobljena podatkovna množica

Označevalcem, študentom 2. stopnje jezikoslovnih smeri, smo predstavili smernice in kriterije za označevanje dveh dimenzij (ne)standardnosti. Vsakemu besedilu so nato pripisali stopnjo standardnosti za obe dimenziji, pri tem pa uporabili vrednost 1 (povsem standardno), 2 (nekoliko nestandardno) ali 3 (zelo nestandardno). Za tristopenjsko lestvico smo se odločili, saj naloga ni (in bi težko bila) zelo natančno definirana.

Po učni fazi, med katero so vsi označevalci označili manjšo množico besedil in prediskutirali rezultate, je označevanje potekalo v dveh kampanjah. V prvem delu je besedila označeval po en označevalec, to pa smo uporabili kot razvojno množico za eksperimente. V drugem delu sta besedila označila po dva označevalca, vzorec pa smo v nadaljevanju uporabili kot testno množico. Končne vrednosti odvisnih spremenljivk za eksperimente smo izračunali kot povprečje vrednosti, ki sta jih pripisala dva označevalca.

2.4 Uporabljene značilke

Za opis tehničnih in jezikovnih značilnosti besedil smo definirali 29 neodvisnih spremenljivk oz. značilk. Za napovedovanje tehnične in jezikovne standardnosti smo uporabili isti nabor značilk. Značilke lahko razdelimo v dve glavni kategoriji.

Značilke na nivoju znakov vključujejo napačno rabo ločil in presledkov, ponavljanje znakov, razmerje med abecednimi in neabecednimi znaki, razmerje med samoglasniki in soglasniki ipd.

Značilke na nivoju pojavnic opisujejo lastnosti besed. Med njimi ločujemo značilke, vezane na niz besed, in značilke, vezane na leksikon, ki temeljijo na zunanjih virih podatkov. Za značilke, vezane na niz besed, smo izračunali delež besed, zapisanih z veliko začetnico ali samimi velikimi črkami, ponovitve besed, delež besed, sestavljenih iz samih soglasnikov, in delež zelo kratkih besed. Za značilke, vezane na leksikon, smo uporabili oblikoslovni leksikon Sloleks (Krek in Erjavec 2009), ki vsebuje slovenske besede z vsemi pregibnimi oblikami. Značilke, ki temeljijo na leksikonu Sloleks, vključujejo delež besed zunaj besedišča, delež besed zunaj besedišča z manjkajočim samoglasnikom, delež kratkih besed zunaj besedišča ipd. Del značilk temelji na korpusu Kres, uravnoteženem korpusu standardne slovenščine (Logar Berginc et al. 2012). Te značilke med drugim vključujejo delež besed zunaj besedišča glede na leksikon pregibnih oblik, ki se v korpusu Kres pojavijo vsaj desetkrat.

2.5 Eksperimenti in rezultati

Za modeliranje odvisnosti med vsako izmed dveh odvisnih spremenljivk (tehnična in jezikovna standardnost) in 29 neodvisnimi spremenljivkami smo testirali številne regresijske modele, s katerimi lahko na podlagi neodvisnih spremenljivk napovemo zvezno vrednost, tj. vrednost med 1 in 3. Na koncu smo se odločili za regresijo z metodo podpornih vektorjev (angl. *SVM regressor*), ki kot jedro uporablja funkcijo RBF (angl. *Radial Basis Function*). Točnost regresorjev smo evalvirali z izračunom povprečne absolutne napake, tj. povprečja razlik med napovedano vrednostjo in vrednostjo, ki so jo pripisali označevalci. Glede na to, da smo računali napako, pomeni nižji rezultat večjo točnost modela. Pri tehnični dimenziji je povprečna absolutna napaka pri najboljšem rezultatu znašala 0,377, pri jezikovni dimenziji pa 0,424, kar nakazuje, da je napovedovanje tehnične standardnosti lažja naloga, vsaj z uporabo našega nabora značilk.

V nadaljevanju smo želeli preveriti, kolikšno izboljšanje modela dosežemo pri uporabi 29 značilk v primerjavi z eno samo, predvsem deležem besed zunaj besedišča, kar je pogost pristop, opisan v literaturi. V ta namen smo zgradili model, ki uporablja samo to značilko. Za tehnično standardnost je povprečna absolutna napaka znašala 0,594, za jezikovno standardnost pa 0,597. Ti rezultati kažejo, da uporaba dodatnih značilk občutno izboljša natančnost modela, predvsem za napovedovanje tehnične standardnosti. Dejstvo, da je ta model boljši pri določanju jezikovne standardnosti, ne preseneča, saj je edina uporabljena značilka (delež besed zunaj besedišča) vezana na jezikovne prvine.

Na koncu smo preverili, kako se naš model razlikuje od osnovnega modela, ki vsakemu besedilu naključno pripiše vrednost med 1 in 3. Za tehnično standardnost znaša povprečna absolutna napaka 0,713, za jezikovno standardnost pa 0,749.

Glede na rezultate lahko sklenemo, da (1) ti še zdaleč niso popolni, vendar (2) so boljši od tistih, ki temeljijo samo na eni značilki (delež besed zunaj besedišča), in (3) veliko boljši od naključnih.

Poleg gradnje modela za napovedovanje standardnosti za slovenščino smo organizirali tudi dodatne označevalske maratone, v okviru katerih smo zbrali podatke za hrvaščino in srbščino, označene na podlagi istih smernic. Fišer et al. (2015) so na podlagi korpusa slovenskih, hrvaških in srbskih tvitov analizirali razlike med napovedano standardnostjo besedil in ugotovili, (1) da so v vseh treh jezikih tviti v veliki večini napisani v standardnem jeziku, (2) da je število nestandardnih besedil najvišje v slovenščini, sledijo jim hrvaška besedila, najmanj pa je srbskih, kar je verjetno posledica različnih stopenj narečne raznolikosti v teh treh jezikih, in da (3) so za slovenske nestandardne tvite značilne predvsem nestandardne pravopisne prvine, medtem ko je nestandardno besedišče pogostejše v hrvaških tvitih, najpogostejše pa v srbskih.

3 SEGMENTACIJA

Delitev besedil na pojavnice (besede in ločila) in stavke na splošno velja za enostavno nalogo. To drži predvsem za jezike, kjer so besede ločene s presledkom, in standardni jezik, kjer so načela za ločevanje besed in stavkov točno določena. Če so ta načela kršena, kar je zelo pogost pojav pri spletnih uporabniških vsebinah, pa postane naloga veliko težja.

3.1 Metoda segmentacije

Za tokenizacijo in stavčno segmentacijo smo razvili orodje v programskem jeziku Python, ki trenutno pokriva slovenščino, hrvaščino in srbščino. Za razliko od ostalih orodij, opisanih v poglavju, ki temeljijo na nadzorovanem strojnem učenju, temelji ta model, kakor večina orodij za segmentacijo, na ročno določenih pravilih, ki so implementirana kot regularni izrazi. Regularni izrazi so definirani na podlagi leksikonov za specifičen jezik, kot so npr. seznami okrajšav. Posebnost tega tokenizatorja je v tem, da ima dva načina: za procesiranje standardnega in nestandardnega jezika.

Način za nestandardni jezik se od standardnega razlikuje v dveh vidikih: (1) definirana pravila so bolj ohlapna kot tista za standardni jezik in (2) dodana so specifična pravila, ki opisujejo pojave, značilne za spletno komunikacijo. Primer takšnega pravila je, da lahko pika konča poved, tudi če se naslednja beseda ne začenja z veliko začetnico ali od pike celo ni ločena s presledkom. Pri tem pa vseeno drži, da se s pojavnico, ki se sicer konča s piko, a je na seznamu okrajšav, ki ne končujejo povedi, npr. *prof.*, poved ne konča. Prav tako je eden izmed dodanih regularnih izrazov za nestandardni način namenjen prepoznavi emotikonov, npr. *:-]*, *:-PPPP*, ^_^ itd.

3.2 Podatkovne množice

Za vse tri jezike smo izvedli evalvacijo nestandardnega načina orodja, pri tem pa smo kot zlati standard uporabili tri podatkovne množice (podrobneje opisane v Čibej et al. 2018), in sicer Janes-Tag 2.0 (Erjavec et al. 2016), ReLDI-NormTagNER-sr 2.0 (Ljubešić et al. 2017a) in ReLDI-NormTagNER-sr 2.0 (Ljubešić et al. 2017a). Poleg drugih nivojev označevanja sta pri teh podatkovnih množicah ročno popravljeni tudi stavčna segmentacija in tokenizacija. Vse tri množice vsebujejo besedila s stopnjami standardnosti T1L1, T1L3, T3L1 in T3L3.

3.3 Eksperimenti in rezultati

Rezultati testiranja orodja na treh podatkovnih množicah so podani v Tabeli 2. Pri računanju natančnosti stavčne segmentacije smo uporabili strogo merilo, in sicer penaliziranje tako v primeru, ko sistem ni ločil povedi, pa bi moral, kot tudi v primeru, ko je izvedel segmentacijo, pa je ne bi smel. Nasprotno smo orodje pri tokenizaciji penalizirali samo za vsako izvirno pojavnico, ki je bila tokenizirana napačno.

Tabela 2: Evalvacija segmentacije.

Jezik	Povedi	Napačnih	Natančnost	Pojavnic	Napačnih	Natančnost
sl	19.009	2.497	86,86 %	184.896	2.067	98,88 %
hr	7.942	1.328	83,28 %	89.208	562	99,37 %
sr	6.902	733	89,38 %	91.853	546	99,41 %

Kot je razvidno iz tabele, znaša natančnost stavčne segmentacije za slovenščino skoraj 87 %, za srbščino je nekoliko višja, za hrvaščino pa nižja. Pri pregledovanju rezultatov za slovenščino se izkaže, da je večina napak posledica povedi, ki se končajo z emotikonom namesto s končnim ločilom, ta pojav pa je izven obsega orodja. Orodje se moti tudi pri nekaterih neprepoznanih okrajšavah, saj končno piko razume kot signal za konec povedi.

Pri tokenizaciji so rezultati najslabši za slovenščino (natančnost znaša tik pod 99 %), najboljši pa ponovno za srbščino z natančnostjo 99,4 %. Pri pregledovanju rezultatov za slovenščino se izkaže, da se največ napak pojavi zaradi vezajev, npr. enota »*nm-lj*«, ki bi morala biti ločena na tri pojavnice. Do napak pride tudi pri napačno tokeniziranih emotikonih in spletnih naslovih – čeprav orodje vsebuje regularne izraze za ti dve vrsti pojavnic, ne pokriva vseh oblik, ki se pojavijo v besedilih. Kot je bilo omenjeno, določene napake povzročijo tudi okrajšave, ki niso zajete v leksikonu orodja.

4 NORMALIZACIJA

Kot smo omenili v uvodu, se za prilagajanje jezikovnih tehnologij za procesiranje nestandardnih besedil uporabljata dva glavna pristopa: (1) normalizacija besedil in uporaba orodij za standardni jezik ter (2) prilagoditev orodij. Prvi pristop je bolj ekonomičen, saj celotnemu postopku zgolj dodamo eno komponento, medtem ko drugi pristop zahteva, da prilagodimo vsak korak procesiranja besedil.

Normalizacija besedil ima še dodatno prednost, ki je pomembna predvsem za (korpusne) jezikoslovce: normalizirani korpus omogoča iskanje besed, ne da bi upoštevali ali sploh poznali vse različice zapisa.

4.1 Metoda normalizacije

Pri projektu JANES smo za normalizacijo besednih oblik uporabili pristop statističnega stojnega prevajanja na nivoju znakov. Pri tem ne gre za prevajanje besed in zvez iz izvornega v ciljni jezik, temveč za pretvorbo znakov in nizov znakov v nestandardnih različicah besed v znake in nize znakov v standardnih različicah.

Model za pretvorbo izvornih nizov znakov v ciljne nize se imenuje *prevodni model*, zgrajen (naučen) pa je na zbirki paralelnih podatkov, tj. obstoječih prevodov ali normalizacij. V paradigmi statističnega strojnega prevajanja obstaja tudi drug zelo uporaben vir informacij, in sicer verjetnostni model različnih nizov v ciljnem jeziku, ki ga imenujemo *jezikovni model*. Glede na to, da je pri normalizaciji ciljni jezik pravzaprav standardni jezik, za gradnjo takšnega jezikovnega modela ni težko zbrati velike količine podatkov. Za slovenščino lahko uporabimo korpus Kres, korpus Gigafida (Logar, 2012) ali slovenski spletni korpus slWaC (Erjavec et al. 2015). Medtem ko se prevodni model uporablja za generiranje hipotez za normalizacijo, se oba modela, skupaj z vrsto dodatnih manjših modelov, uporabljata za identifikacijo najbolj verjetne normalizirane oblike.

Primer iz nestandardne slovenščine, ki se ga model v tej paradigmi nauči zelo hitro, je nestandardna končnica pri deležnikih na *-l* določenih glagolov (*naredil* vs. *naredu*). Če imamo v učni množici primere takšnih transformacij (besede, ki se končajo na »du«, pretvorjene v besede, ki se končajo na »dil«), lahko prevodni model zlahka generira hipotezo *pobegnil* za nestandardno obliko *pobegnu* in ji pripiše precej visoko verjetnost, čeprav te pojavnice v učni množici še ni videl. Jezikovni model lahko pripisano verjetnost še poveča, saj je verjetnost za pojavitev niza znakov *pobegnil* v standardni slovenščini zelo velika in veliko večja kot verjetnost za niz *pobegnu* ali katero drugo možno hipotezo.

Prednost pristopov nadzorovanega strojnega učenja je v tem, da lahko isti učni algoritem uporabimo za reševanje podobnih problemov, če le imamo na voljo učne podatke. Med eksperimenti smo se poleg normalizacije spletnih uporabniških vsebin ukvarjali tudi s prevajanjem zgodovinskih besedil v sodobni jezik. Drugi problemi, ki bi jih lahko reševali na podoben način, vključujejo prevajanje med narečji, odpravljanje pravopisnih in slovničnih napak pa tudi popravljanje napak, ki jih napravijo osebe z različnimi jezikovnimi motnjami.

V nadaljevanju predstavimo rezultate eksperimentov, ki so podrobno opisani v Ljubešić et al. (2016). V članku smo identificirali optimalni način za uporabo strojnega prevajanja na nivoju znakov tako pri normalizaciji spletnih uporabniških vsebin kot pri prevajanju zgodovinskih besedil v sodobni jezik. Orodje, ki smo ga uporabili za slovenščino, smo uporabili tudi pri eksperimentih za prevajanje narečne švicarske nemščine v metašvicarsko nemščino, ki se precej približa standardni, opisanih v Scherrer in Ljubešić (2016).

4.2 Podatkovne množice

Eksperimente smo izvedli na podatkih iz zgodnje različice korpusa Janes-Norm (glej Čibej et al. 2018), ki je vseboval 1.000 tvitov, označenih kot povsem standardnih (L1), in 1.000 tvitov, označenih kot zelo nestandardnih (L3). Tako je podatkovna množica za kategorijo spletnih uporabniških vsebin vključevala za normalizacijo težjo množico podatkov L3, kot tudi lažjo množico podatkov L1.

Dodatno smo izvedli eksperimente z zgodovinskimi besedili, kjer smo uporabili ročno označeni korpus starejše slovenščine goo300k (Erjavec, 2015), ki vključuje transkripcije 1.100 strani besedil (približno 300.000 pojavnic), vzorčenih iz 88 knjig in enega časopisa, izdanih med letoma 1584 in 1899. Vsaki pojavnici v korpusu je pripisana normalizirana (sodobna) besedna oblika. Za potrebe eksperimenta korpus ločimo na dva dela: težjo in lažjo množico podatkov, pri čemer težja vsebuje besedila, napisana v bohoričici, ki se precej razlikuje od sodobnega jezika, lažja pa gajici in je bližje sodobnemu jeziku.

4.3 Eksperimenti in rezultati

Z eksperimenti smo skušali odgovoriti na dve glavni raziskovalni vprašanji: (1) ali obstaja en sam model statističnega strojnega prevajanja na nivoju znakov, ki je najbolj učinkovit za normalizacijo ne glede na to, ali normaliziramo spletne uporabniške vsebine ali zgodovinska besedila, in (2) ali lahko izboljšamo tradicionalno normalizacijo po pojavnicah tako, da uporabimo prevajanje celotnih segmentov in s tem upoštevamo kontekst.

Drugo vprašanje se nanaša na dejstvo, da večina sodobnih pristopov temelji na normalizaciji na nivoju pojavnic, kar pomeni, da ne upoštevajo konteksta, v katerem se besede pojavljajo. Tovrstni pristopi ne sledijo jezikovni intuiciji o pomembnosti konteksta za ustrezno normalizacijo.

Za evalvacijo modela smo uporabili prilagojeno Levenshteinovo razdaljo, tj. odstotek znakov, ki bi jih morali zamenjati, da bi bila normalizacija identična referenčni normalizaciji. Kot referenco smo uporabili mero *Leave-As-Is* (LAI), tj. proces, ki vhodnih podatkov ne spremeni. Razlog za uporabo te reference je v tem, da lahko za določeno podatkovno množico tako izmerimo kompleksnost problema in dodani odstotek primerov, ki smo jih uspeli razrešiti z določenim pristopom. Rezultati eksperimentov so prikazani v Tabeli 3.

Tabela 3: Vrednosti relativne Levenshteinove razdalje za normalizacijo težjih in lažjih primerov pri spletnih uporabniških vsebinah in pri starejši slovenščini.

	LAI	pojavnica	segment	+JM pojavnica	+JM segment
L3	5,15	2,19	2,12	1,76	1,58
L1	0,75	0,41	0,43	0,34	0,38
Bohorič	17,63	1,55	1,92	1,51	1,33
Gaj	3,13	1,01	1,15	0,91	0,93

V prvem stolpcu so prikazane podatkovne množice, v drugem stolpcu pa mera LAI, ki kaže, kako daleč je določena podatkovna množica od standardizirane različice. Pri spletnih uporabniških vsebinah mora biti za manj standardno množico (L3) popravljenih 5 % znakov, za bolj standardno množico (L1) pa manj kot 1 %. Pri zgodovinskih besedilih mora biti za množico z bohoričico spremenjenih 18 % znakov, pri množici z gajico pa 3 % znakov.

Učenje normalizacije na nivoju pojavnic (stolpec *pojavnica*) je v primerjavi z referenco (LAI) zelo uspešno. Pri najmanj standardnih podatkih (bohoričica) je kljub relativno majhni učni množici napačnih zgolj 9 % transformacij, 91 % primerov pa je že razrešenih (relativna Levenshteinova razdalja se s 17,63 zniža na 1,55). Pri preostalih treh podatkovnih množicah število napak prav tako pomembno upade, čeprav v manjši meri. Zanimivo je, da je prevajanje celotnih segmentov (stolpec *segment*) v večini primerov manj uspešno kot normalizacija posameznih pojavnic. Razlog za to bi lahko bil v tem, da sta učna množica in jezikovni model precej majhna.

Zadnja dva stolpca prikazujeta rezultate, kjer smo uporabili jezikovne modele, naučene na velikih zbirkah večinoma standardnih, sodobnih podatkov. Prva ugotovitev je, da uporaba dodatnih jezikovnih modelov izboljša rezultate pri obeh nalogah. Še zanimivejša je ugotovitev, da je pri težjih množicah podatkov (L3 in bohoričica) normalizacija na nivoju segmentov bolj uspešna kot normalizacija na nivoju pojavnic. Odstotek primerov, ki so bili razrešeni, se giblje med 49 % in 92 %.

Rezultate za slovenščino smo primerjali z rezultati za narečno švicarsko nemščino, opisanimi v Scherrer in Ljubešić (2016). Pri uporabi normalizacije celotnih povedi namesto posameznih pojavnic se je pri tej nalogi število napak zmanjšalo za 20 %. Pri podatkovni množici z bohoričico se je število napak pri istem scenariju zmanjšalo za 12 %, pri podatkovni množici L3 pa za 10 %. Nasprotno se je za podatkovno množico z gajico in podatkovno množico L1 pri prevajanju celotnih povedi število napak povečalo. Pregledali smo vse podatkovne množice in predlagali metriko, izračunano na pojavnicah v izvornem jeziku, ki bi lahko pokazala, ali bi normalizacija na nivoju segmentov izboljšala rezultate: izračunali smo število pojavnic, ki imajo več možnih normalizacij, izbrana pa mora biti manj pogosta. Na ta način smo izmerili odstotek pojavnic, pri katerih je za pravilno normalizacijo nujen kontekst, saj bi bila pri normalizaciji na nivoju pojavnic izbrana napačna, najpogostejša normalizacija. Pri švicarski množici je takšnih pojavnic 7 %, medtem ko je pri množici z bohoričico 5,5 %, pri množici z gajico 3,5 %, pri množici L3 6,1 % in pri množici L1 1,6 % pojavnic, ki za pravilno normalizacijo zahtevajo kontekst. Definirali smo hevristiko, ki predvideva uporabo normalizacije celotnih povedi v primeru, da več kot 4 % pojavnic ne moremo pravilno normalizirati brez konteksta.

Najbolj uspešen sistem za vse opisane eksperimente je bil objavljen kot orodje, imenovano *csmtiser*. S tem orodjem smo sodelovali tudi na tekmovanju CLIN2017 v prevajanju starejše nizozemščine v sodobno nizozemščino, kjer smo med 8 evropskimi univerzami pri normalizaciji dosegli najboljši rezultat (Tjong Kim Sang et al. 2017).

5 REDIAKRITIZACIJA

Pri računalniško posredovani komunikaciji uporabniki, ki pišejo v latinici, znake s strešicami iz ergonomskih razlogov pogosto zamenjajo z ustreznicami iz nabora ASCII, predvsem pri tipkanju na tablicah ali pametnih telefonih. Branje takšnih besedil ljudem praviloma ne povzroča težav, računalniško procesiranje pa je zelo zahtevno, saj je veliko besed brez šumnikov neznanih ali dvoumnih. Iz tega razloga je rediakritizacija besedil zelo aktivno raziskovalno področje. Razvili smo orodje za avtomatsko rediakritizacijo (Ljubešić et al. 2016), ki smo ga naučili in testirali na slovenskih, hrvaških in srbskih besedilih (srbska besedila so bila napisana v latinici). Tega orodja za končno normalizacijo korpusa Janes nismo uporabili (za simultano rediakritizacijo in standardizacijo smo uporabili orodje *csmtizer*, glej razdelek 4), vseeno pa je v nadaljevanju podrobneje opisano, saj je uporabno za kakovostno rediakritizacijo, računalniško manj kompleksno in zato tudi hitrejše, prav tako pa ga je lažje namestiti kot celotno normalizacijsko orodje.

5.1 Podatkovne množice

Za učenje modela smo za vse tri jezike uporabili tri vrste besedil: besedila z Wikipedije, spletna besedila in nestandardna besedila s Twitterja. Ker smo želeli, da orodje pokriva tako besedila, napisana v standardnem jeziku, kot tista, ki so pogosto nestandardna, smo za standardni nabor kot testno množico uporabili besedila z Wikipedije, za nestandardni pa besedila s Twitterja. Korpuse za Wikipedijo smo zgradili s pomočjo splošne skripte za zajem besedil z Wikipedije. Obdržali smo samo povedi, ki vsebujejo 100 znakov ali več, in na ta način odstranili večino preostalih šumnih podatkov. Podatkovna množica za slovenščino je vsebovala približno 20 milijonov, za hrvaščino 28 milijonov in za srbščino skoraj 34 milijonov besed.

Za gradnjo spletnih korpusov smo uporabili korpuse WaC za tri obravnavane jezike (Ljubešić in Klubička 2014; Erjavec in Ljubešić 2014). Ker smo želeli pridobiti dobro učno množico, besedila na spletu pa so lahko napisana tudi brez strešic, smo vključili zgolj tista besedila, pri katerih vsaj 20 % pojavnic vsebuje diakritična znamenja. Čeprav gre za precej strogo merilo, na podlagi katerega smo izključili tudi besedila s pravilno rabo strešic, je spletnih besedil toliko, da so bile dobljene podatkovne množice vseeno zelo velike. Podatkovna množica s spletnimi besedili vsebuje za slovenščino več kot 130 milijonov, za hrvaščino skoraj 270 milijonov in za srbščino 103 milijone besed. Uporabljeno merilo z 20 % šumnikov je bilo definirano na podlagi ročnega pregledovanja podatkov in z glavnim ciljem zagotoviti čim večjo natančnost modela.

Za gradnjo korpusov s Twitterja smo uporabili veliko zbirko tvitov, ki smo jih zbirali od sredine leta 2013 z orodjem TweetCat (Ljubešić et al. 2014). Ker sta hrvaščina in srbščina zelo podobna jezika, smo za razlikovanje med hrvaškimi in srbskimi uporabniki uporabili namensko razvito orodje (Ljubešić in Kranjčić 2015). Ker smo želeli, da podatkovne množice s Twitterja vsebujejo predvsem nestandardna besedila, smo vključili samo tiste tvite, ki so bili avtomatsko označeni kot nekoliko ali zelo jezikovno nestandardni (L2 in L3). Na koncu smo izključili še vse tvite, pri katerih manj kot 10 % pojavnic vsebuje diakritična znamenja. V tem primeru smo uporabili manj strogo merilo kot pri spletnih besedilih, saj je število podatkov s Twitterja veliko nižje. Podatkovna množica za slovenščino vsebuje le 6,7 milijona, za hrvaščino 1,9 milijona in za srbščino 13,7 milijona besed. Merilo (10 %) je bilo tako kot prej definirano na podlagi ročnega pregledovanja podatkov, glavna cilja pa sta bila visoka natančnost in priklic.

Podatki z Wikipedije so bili ločeni na stavke in pojavnice, pri podatkih s Twitterja pa smo kot osnovno enoto uporabili celoten tvit. Prav tako smo za vse pojavnice uporabili zapis z malimi črkami in na ta način zmanjšali razpršenost podatkov. Pri

začetnih eksperimentih se je namreč izkazalo, da ohranitev velikih in malih črk za obravnavane jezike nima informativne vrednosti. Pretvorba pojavnic v prvotni zapis z velikimi in malimi črkami po rediakritizaciji ne predstavlja težav, saj se v večini primerov število črk v pojavnici ne spremeni.

Podatke smo razdelili na učne, razvojne in testne množice in iz učnih množic odstranili vse duplikate. Da bi sistem prilagodili za standardne in nestandardne podatke, smo iz zbirk besedil z Wikipedije in Twitterja izločili razvojne množice, ki so vsebovale po 10.000 besedil za vsako vrsto besedila in jezik. Za testiranje modela na standardnih podatkih smo uporabili dodatnih 10.000 besedil iz podatkovnih množic za Wikipedijo. Glede na to, da so med filtriranimi podatki lahko napake oz. izpusti strešic, smo za testiranje na nestandardnih podatkih za vsak jezik izdelali testno množico po 2.000 tvitov, ki so jih pregledali jezikoslovci. Te tvite smo zajeli iz prvotne množice tvitov, vendar se pri njih nismo držali omejitve števila pojavnic, ki vsebujejo diakritična znamenja, saj smo želeli zagotoviti, da bo testna množica reprezentativna za nestandardni jezik na splošno. Iz razvojnih in testnih množic prav tako nismo odstranili duplikatov.

5.2 Eksperimenti in rezultati

Rediakritizacijo smo definirali kot prevajalski problem na nivoju pojavnic, kjer je vsaka pojavnica »prevedena« v različico s pravilno postavljenimi strešicami na šumnikih. Za učenje sistema smo iz izvornih besedil preprosto odstranili strešice in tako ustvarili paralelno podatkovno množico, poravnano na nivoju pojavnic. Za reševanje tega prevajalskega problema smo uporabili dva pristopa:

- *leksikonski pristop* (leksikon) – uporaba najpogostejšega prevoda iz učne množice
- *korpusni pristop* (TM+LM) – kombiniranje informacij o tem, kako verjeten je prevod (*translation model* – TM) in kako verjetna je pojavitev prevoda v določenem kontekstu (*language model* – LM), z uporabo log-linearnega modela, ocenjenega na podlagi razvojne množice

Pri obeh pristopih smo za ocenjevanje verjetnosti prevoda uporabili oceno največje verjetnosti (angl. *maximum likelihood estimate*) za obliko s strešicami glede na obliko brez strešic. Za ocenjevanje kontekstualne verjetnosti smo uporabili dobro poznano orodje za modeliranje jezika KenLM (Heafield, 2011) s privzetimi parametri.

Za ocenjevanje dveh parametrov log-linearnega modela smo izvedli izčrpno iskanje med vsemi kombinacijami v intervalu [0,0, 1,0] s korakom 0,1. Kot ciljno

funkcijo smo uporabili točnost pojavnice. Za glajenje podatkov v preiskovalnem prostoru smo za vsako kombinacijo parametrov povprečili rezultate za en korak večjih in manjših vrednosti parametra. Tako smo za kombinacijo parametrov (0,2, 0,3) vzeli povprečje meritev naslednjih kombinacij: (0,1, 0,3), (0,3, 0,3), (0,2, 0,3), (0,2, 0,2) in (0,2, 0,4).

Tabela 4 prikazuje rezultate najuspešnejšega pristopa, in sicer metode TM+LM, naučene na vseh podatkih, tj. skupku podatkov z Wikipedije, Twitterja in spleta. Podatki označujejo točnost na besedo, ne glede na to, ali je ta beseda kandidat za rediakritizacijo ali ne. Kot je razvidno iz preglednice, so rezultati precej dobri; v vseh primerih je stopnja napake manjša kot 1 %. Rezultati prav tako kažejo, da naše korpusne metode delujejo veliko bolje kot edino prosto dostopno orodje za rediakritizacijo, imenovano *charlifter*.[1]

Na splošno se testna množica z Wikipedije izkaže za najenostavnejšo in za slovenščino doseže najboljše rezultate, medtem ko za hrvaščino najboljše rezultate dosežejo podatki s Twitterja. V zadnji vrstici je prikazano, kolikšno zmanjšanje napake zagotovi naša metoda v primerjavi z enostavno metodo z leksikonom, kjer za vsako besedo z morebitnimi diakritičnimi znamenji poiščemo besedo v Sloleksu in ji v skladu s tem dodamo strešice. Pri naši metodi stopnja napake znatno upade, in sicer zagotovi od skoraj četrtine do več kot polovice manj napak.

Tabela 4: Rezultati uporabe orodja za rediakritizacijo z najboljšim modelom za različne podatkovne množice in jezike. Zadnja vrstica prikazuje zmanjšanje števila napak v primerjavi z enostavno metodo z leksikonom.

	Wiki-sl	Wiki-hr	Wiki-sr	Tweet-sl	Tweet-hr	Tweet-sr
brez intervencije	0,8615	0,8614	0,8844	0,8715	0,8397	0,8730
točnost *charlifter*	0,9790	0,9674	0,9706	0,9508	0,9436	0,9330
točnost TM+LM	0,9962	0,9957	0,9947	0,9912	0,9938	0,9917
Δ metode z leksikonom	32,81%	30,26%	43,28%	25,30%	22,43%	51,11%

Da bi dobili vpogled v naravo napak, ki se pojavljajo pri najuspešnejšem modelu, smo opravili tudi analizo napak za podatkovni množici z besedili z Wikipedije in Twitterja v slovenščini. V vsaki podatkovni množici smo ročno pregledali 100 napak in jih razvrstili v 9 kategorij, ki so opisane v Tabeli 5.

1 https://sourceforge.net/projects/lingala/files/charlifter/

Tabela 5: Rezultati ročne analize napak za slovenščino s primeri.

Tip napake	Wikipedija	Twitter	Primeri
lastno ime	30	6	*petar *šegvić*, mesto *kiš**
redka beseda	28	6	*osemkotno *užleblјenje*, *šamaševa* tablica*
dvoumna beseda	21	37	*šoja / soja, teza / teža*
tuja beseda	8	3	*DE *das* antlitz der erde, kaj potegnit za your *case**
tipkarska napaka	6	6	*naj *počakojo* nasprotnika, operacijski *ojačvealniki**
težava s tokenizacijo	4	31	*zaostajali ali *bilispuščeni* → bili spuščeni, Wiki: en - *sipad* - zid - ana *laraški* → en-sipad-zin-ana laraški*
pravilna različica	3	3	*inštitucija / institucija, špirala / spirala*
ponovljene črke	0	5	*kolk sem *žiivčna*, *sončči* *špeeegliiii**
napaka testne množice	0	3	*član los *angaleske* skupine → član losangeleške skupine, pa *se* hipster si → pa še hipster si*
	100	100	

Rezultati analize kažejo, da se razlogi za napake v obeh podatkovnih množicah precej razlikujejo. Pri Wikipediji največjo težavo povzročajo lastna imena, ki jih v učni množici ni bilo (30 %, npr. *japonski umetnik Hirošige, hrvaški pevec Vinko Coce, sumersko mesto Ešnuna, Jangončani – prebivalci burmanskega mesta Jangon*), in redke besede, značilne za specifično domeno, pogosto izpeljane iz tujih besed ali lastnih imen (28 %, npr. *senponški škof, pižanski koncil, komodoški varan, Jastrebova stela*). Pri tvitih sta glavna razloga za napačno rediakritizacijo dvoumnost besed – besede, ki obstajajo s strešicami in brez njih (37 %, npr. *se/še, reci/reči, car/čar, nas/naš, poklice/pokliče*), in izpuščanje presledkov (31 %, npr. *splohnisemnatekocem, #sanjskikrozek*) bodisi za varčevanje s prostorom in časom bodisi kot pogost fenomen pri ključnikih. Najhujše napake se pojavijo pri dvoumnih besedah (21 % pri standardnih besedilih in 37 % pri nestandardnih besedilih), zato bi morali v prihodnje največ pozornosti nameniti odpravljanju teh napak.

6 OBLIKOSKLADENJSKO OZNAČEVANJE

Prednosti normalizacije nestandardnih podatkov in uporabe standardnega pristopa procesiranja besedil so bile opisane, v naslednjih dveh poglavjih pa podamo

še argumente za prilagoditev orodij za nestandardni jezik. Najpomembnejši argument za neposredno procesiranje nestandardnega jezika je ta, da s predhodno normalizacijo izgubimo določene informacije in zato kasnejša označevanja ne morejo delovati tako dobro kot z uporabo modelov, naučenih na nestandardnih podatkih. Vsakršno avtomatsko procesiranje prav tako pripelje do napak, ki se prenesejo v nadaljnje faze procesiranja, hkrati pa lahko negativno vplivajo na procesiranje sosednjih ali kako drugače povezanih pojavnic. Prilagajanje orodij za nestandardni jezik prav tako ni nujno zelo potratno, saj že majhna učna množica domensko specifičnih/nestandardnih besedil zadošča, da se sistem nauči vsaj zelo pogostih fenomenov.

V tem razdelku predstavimo pristop prilagajanja najsodobnejšega označevalnika slovenskih (Ljubešić in Erjavec, 2016), hrvaških in srbskih besedil (Ljubešić et al. 2016a) za nestandardni jezik s primerom za slovenščino (Ljubešić et al. 2017). Vse metode temeljijo na učenju pogojnih naključnih polj, tj. metodi za učenje zaporednega označevanja.

Izvedli smo dve vrsti prilagoditev: (1) z vključitvijo nestandardnih učnih podatkov (nadzorovana prilagoditev) in (2) z vključitvijo dodatnih informacij, naučenih iz velikih zbirk surovih nestandardnih podatkov, v obliki Brownovih gruč (nenadzorovana prilagoditev).

Brownovo gručenje (Brown et al. 1992) je tehnika hierarhičnega gručenja besed, tj. procesa razvrščanja besed v skupine glede na podobnost konteksta, v katerem se pojavljajo. Ta nenadzorovana metoda (ne zahteva ročno označenih podatkov) je pri procesiranju naravnega jezika zelo priljubljena, saj predstavlja zelo poceni način prilagajanja orodij za različne domene.

V Tabeli 6 so prikazani rezultati Brownovega gručenja za slovenski spletni korpus. Prva gruča vsebuje niz nedoločnikov, med katerimi so nekateri zapisani v nestandardni okrajšani obliki brez končnega *-i*. Z informacijo, da je določena beseda v tej gruči, označevalniku omogočimo, da se nauči odvisnosti med oznako za glagolsko nedoločnost in identifikatorjem te gruče. Na ta način lahko označevalnik na podlagi konteksta in informacije o gruči za fenomen, ki ga še ni videl, npr. okrajšano različico nedoločnika, napove pravilno oznako. Druga gruča vsebuje različne oblike črkovanja za osebni zaimek, tretja in četrta pa standardne in nestandardne oblike prislovov. Kot lahko vidimo, vsebuje četrta gruča tudi oblike, ki niso prislovi, kar pomeni, da s to metodo pridobimo zgolj približne rezultate. Vseeno pa te informacije kljub šumnim podatkom pripomorejo k izboljšanju orodij za procesiranje naravnega jezika.

Tabela 6: Primeri Brownovih gruč, izračunanih iz slovenskega spletnega korpusa.

narediti storiti naučiti napisati početi izgubiti naredit napraviti poskusiti zaslužiti pojesti obleči tvegati plačat postoriti shujšati žrtvovati narest prebrat popiti zamenjat nardit rešit skuhati počet spremenit zapraviti popravit potrpeti privarčevati poizkusiti pojest spiti menjat naučit ukreniti požreti prodat izmisliti zmenit nastavit dodat pripravit uredit
jaz jst jest js jz
marsikaj karkoli kej kj karkol kàj
itak tud kr tut kao skoz ziher tle lahk skor tm zdele prov valda tuki skos dons zihr lohk una nonstop edin dans prou loh itaq napisu non-stop valjda kle počas ponavad već kmal nardil tamo clo nešto prec čak tukej opet lohka ratala verjetn

6.1 Podatkovne množice

Za nadzorovano prilagoditev orodij smo uporabili podatkovno množico Janes-Tag (glej poglavje Čibej et al. 2018). Za namene eksperimentov smo podatke razdelili na deleže 80 : 10 : 10, ki so služili kot učna, razvojna in testna množica.

Za računanje Brownovih gruč smo uporabili (1) 1,2 milijarde pojavnic iz spletnega korpusa za slovenščino slWaC v2.0 (Erjavec et al. 2015), (2) korpus Janes v.04 in (3) skupek obeh korpusov. Za vsak vir smo besede, ki se pojavijo vsaj 50-krat, združili v 2.000 gruč.

Na koncu smo preverili tudi, kako se rezultati spremenijo, če v nabor značilk vključimo normalizirane oblike. Za normalizacijo smo uporabili delež podatkovne množice Janes-Norm (opisana v poglavju Čibej et al. 2018), ki se ne prekriva z množico Janes-Tag, saj smo želeli zagotoviti, da učenje normalizatorja ne bo potekalo na istih podatkih, ki jih mora kasneje normalizirati.

6.2 Značilke

Osnovne značilke, ki smo jih uporabili za označevalnik, so naslednje: pojavnice, zapisane z malimi črkami, na položajih {-3, -2 ... 3}; pripone obravnavane pojavnice (pojavnica na položaju 0, ki jo sistem trenutno označuje) dolžine {1, 2, 3, 4}; hipoteze za oznako, ki jih pridobimo iz oblikoslovnega leksikona, za pojavnice na položajih {-2, -1 ... 2} in predstavitve obravnavanih pojavnic, ki zaznamujejo, ali pojavnica vsebuje številke, velike črke, male črke ali druge simbole (npr. pojavnica *Gr8t* je zaznamovana z »uldl«, saj je sestavljena iz velike

črke (*upper*), male črke (*lower*), številke (*digit*) in male črke), in ali se pojavi na začetku povedi.

Za dodajanje informacij o Brownovem gručenju smo v skupino značilk vključili različne informacije o gručah, npr. celotni identifikator gruč in binarne poti različnih dolžin v binarnem drevesu, in na ta način zagotovili informacije o gručenje z različnimi nivoji podrobnosti.

Na koncu smo v nabor značilk dodali tudi podatke o normalizaciji, in sicer hipoteze za ustrezno oblikoskladenjsko oznako, izračunane iz oblikoslovnega leksikona na podlagi normalizirane oblike pojavnice.

6.3 Eksperimenti in rezultati

S prvim eksperimentom smo merili, kako se stopnja napake poveča, če s standardnim modelom označimo nestandardne podatke. Za standardno testno množico je bila točnost (odstotek pravilno označenih pojavnic) 94-%, pri nestandardnih podatkih pa je model dosegel le 69-% točnost. Nadzorovana prilagoditev označevalnika z učenjem na podatkovni množici Janes-Tag je točnost zvišala na 84 %. Pri kombiniranju standardne učne množice in množice Janes-Tag se je točnost še povišala, in sicer na 86 %.

Z nadaljnjimi eksperimenti smo preverjali, kako dodajanje Brownovih gruč v nabor značilk vpliva na točnost modela. Za te eksperimente smo uporabili sistem, naučen zgolj na podatkih iz množice Janes-Tag. Z dodajanjem informacij o gručenju se je prvotna točnost modela (84 %) zvišala na 86 %. Dodajanje značilk, ki smo jih pridobili iz normaliziranih oblik besed, je rezultate izboljšalo samo za pol odstotka, v primeru, da bi bile na voljo popolne normalizacije besed, pa bi točnost znašala 88 %.

Z združevanjem standardne in nestandardne učne množice in uporabo dodatnih značilk z informacijami o Brownovem gručenju in avtomatski normalizaciji smo dosegli najboljši rezultat, in sicer 88-% točnost. Medtem ko je ta rezultat v primerjavi s prvotno 69-% točnostjo veliko boljši, je še vseeno precej daleč od točnosti za standardna besedila, ki znaša 94 %.

7 RAZPOZNAVANJE IMENSKIH ENTITET

Zadnje orodje, ki ga opišemo v tem poglavju, je tudi zadnje, ki smo ga razvili v okviru projekta JANES. Označevanje imenskih entitet je v splošnem zelo

uporabno, pri projektu pa je bil glavni namen za razvoj tega orodja avtomatska anonimizacija besedil, potrebna za objavo zgrajenih korpusov.

7.1 Metoda

Orodje je zelo podobno oblikoskladenjskemu označevalniku, saj vključuje iste osnovne značilke (1) brez hipotez na podlagi oblikoslovnega leksikona, (2) z značilkami glede Brownovega gručenja in (3) z dvema dodatnima značilkama, ki opisujeta napovedano besedno vrsto in celotno oblikoskladenjsko oznako.

7.2 Podatkovne množice

Za učenje orodja smo uporabili delež nove različice korpusa ssj500k (Krek et al. 2015), ki vsebuje oznake imenskih entitet, in podatkovne množice Janes-Tag (opisana v Čibej et al. 2018), ki je bil označen na enak način kot nova različica ssj500k. Imenske entitete v obeh korpusih so klasificirane v osebna imena (*oseba*), svojilne pridevnike, izpeljane iz osebnega imena (*izpeljano iz osebe*), krajevna imena (*lokacija*), imena organizacij (*organizacija*) in druga imena (*drugo*).

7.3 Eksperimenti in rezultati

Orodje je bilo izdelano na podlagi številnih predhodnih izkušenj tako v označevanju zaporedij (Ljubešić in Erjavec, 2016; Ljubešić et al. 2016a) kot tudi v razpoznavanju imenskih entitet (Ljubešić et al. 2013), tako da med izdelavo orodja nismo izvedli obsežnejših eksperimentov. Evalvacijo smo opravili naknadno, pri tem pa 80 % podatkov uporabili za učenje in 20 % za testiranje. Glede na to, da podatkovna množica vsebuje standardna in nestandardna besedila, v nadaljevanju poročamo o rezultatih, ki smo jih dosegli pri uporabi (1) samo standardnih podatkov, (2) samo nestandardnih podatkov in (3) kombinacije standardnih in nestandardnih podatkov. Dodatno smo izvedli tudi eksperiment (4), pri katerem smo model učili na standardnih, testirali pa na nestandardnih podatkih. Rezultati evalvacije so podani v Tabeli 7. Za vsako kategorijo so podani rezultati za natančnost, priklic in F1 – harmonično povprečje natančnosti in priklica.

Rezultati prvih treh eksperimentov kažejo, da je razpoznavanje imenskih entitet najuspešnejše pri kategoriji *oseba*. Kot je bilo pričakovano, je na drugem mestu kategorija *lokacija*, sledita pa ji kategorija *organizacija* in na koncu kategorija

drugo. Rezultati za novo kategorijo *izpeljano iz osebe*, ki do sedaj za slovenščino še ni bila vključena, so slabši, kot bi pričakovali, saj ima večina izpeljank obliko svojilnih pridevnikov s točno določeno končnico. Ta kategorija se v nestandardni testni množici pojavi zgolj enkrat in ima zato pri teh pogojih slabe rezultate. Pri primerjavi standardnih in nestandardnih besedil opazimo, da boljši rezultat pri nestandardnih besedilih dosega kategorija *oseba*, in sicer zaradi omemb uporabniških imen, ki se začenjajo z @ in se jih zato model zlahka nauči. Pri kategoriji *lokacija* so rezultati primerljivi, kategorija *organizacija* pa ima pri nestandardnih besedilih veliko slabši rezultat.

Rezultati četrtega eksperimenta, učenja modela na standardnih podatkih in testiranja na nestandardnih, izkazujejo znaten upad F1 pri vseh kategorijah, predvsem pri *organizaciji* in *osebi*, kar je podobno kot pri oblikoskladenjskem označevanju. Vzrok za drastični upad pri kategoriji *organizacija* so različni načini označevanja teh entitet v obeh vrstah besedil, upad pri kategoriji *oseba* pa je po vsej verjetnosti posledica pogostih omemb uporabniških imen, ki jih v standardni učni množici ne srečamo.

Tabela 7: Rezultati evalvacije (natančnost, priklic, F1) razpoznavanja imenskih entitet.

	Standardno			Nestandardno			Oboje			Standardno na nestandardnem		
oseba	0,87	0,95	0,91	0,98	1,00	0,99	0,88	0,96	0,92	0,89	0,20	0,34
izpelj.	0,44	0,56	0,49	0,00	0,00	0,00	0,44	0,52	0,48	0,00	0,00	0,00
lokac.	0,85	0,74	0,79	0,79	0,92	0,85	0,85	0,75	0,80	0,57	0,57	0,62
organ.	0,69	0,48	0,57	0,50	0,33	0,40	0,69	0,48	0,56	0,00	0,00	0,00
drugo	0,39	0,24	0,30	0,75	0,21	0,33	0,41	0,24	0,30	0,60	0,21	0,22

Pri standardnih podatkih smo prekosili do sedaj najboljše rezultate za standardno slovenščino (Štajner et al. 2013), kjer F1 za osebna imena znaša 0,84, za zemljepisna imena 0,76 in imena organizacij 0,56, prav tako pa smo kot prvi predstavili rezultate za nestandardno slovenščino.

8 SKLEP

V poglavju smo opisali skupino orodij za procesiranje nestandardnih besedil v slovenščini, ki smo jih razvili v okviru projekta JANES. Pokazali smo, da lahko dosežemo velika izboljšanja pri jezikovnem procesiranju nestandardnih besedil. Največje izboljšave pogosto dosežemo že z majhno množico domenskih podatkov, npr. nestandardnih podatkov, označenih glede na poljubni fenomen.

Vsa predstavljena orodja (in druga) so prosto dostopna v GitHub repozitoriju CLARIN.SI,[2] kar omogoča, da drugi raziskovalci ne le uporabljajo, ampak tudi prispevajo k orodjem, s tem da poročajo o težavah ali iz obstoječih modelov zgradijo lastne, izboljšane različice orodij. Izbrana orodja za označevanje smo vključili tudi v spletno okolje za gradnjo delotokov ClowdFlows (opisano v poglavju Martinc et al. 2018), prav tako pa jih nameravamo vključiti v podobno okolje WebLicht (Ljubešić et al. 2017), ki ga je razvil CLARIN-DE (Hinrichs et al. 2010).

Najsodobnejše jezikovne tehnologije temeljijo na paradigmi nadzorovanega strojnega učenja in zdi se, da se to v prihodnosti ne bo spremenilo. Strojno učenje in s tem povezana področja (npr. procesiranje naravnega jezika) se namreč nagibajo h globokemu učenju, tj. uporabi nevronskih mrež. Pri procesiranju zveznih signalov (npr. zvok, slika, video) uporaba globokega učenja znatno zmanjša število napak v primerjavi s prejšnjimi pristopi, prav tako globoko učenje izboljša jezikovno procesiranje besedil, čeprav v precej manjši meri. *Bilty* (Plank et al. 2016), najsodobnejši označevalnik, ki temelji na globokem učenju, dosega primerljive rezultate kot označevalnik za slovenščino, ki smo ga opisali v tem poglavju. Največja razlika med tradicionalnimi pristopi, opisanimi v tem poglavju, in nevronskimi mrežami pa je v tem, da nevronske mreže ne zahtevajo oblikovanja značilk, temveč relevantne dele signala za določeno nalogo identificirajo same. Tako bo priprava orodij za procesiranje jezika v bližnji prihodnosti vključevala (1) uporabo sodobnega orodja in učenje na podatkovni množici ali (2) razvoj posebnega orodja, ki bo primarno vključeval definiranje arhitekture nevronske mreže, in učenje nevronske mreže na dani podatkovni množici. Glede na to, da se obseg dela razvijalcev orodij manjša, potreba po velikih količinah kakovostnih ročno označenih podatkov pa vztrajno narašča (nevronske mreže so uspešne predvsem pri zelo velikih količinah podatkov), zagovarjamo vse večjo pomembnost strokovnjakov s področij jezikovnih tehnologij (jezikoslovje, obdelava podatkov) in sorodnih disciplin digitalne humanistike in družbenih znanosti, na primer družbenih tehnologij (prediktorji sociodemografskih spremenljivk za določene govorce ipd.). Izdelava kakovostno označenih podatkovnih množic, kot smo jih uporabili pri gradnji orodij, opisanih v tem poglavju, predstavlja kompleksno in drago nalogo, ki pa bo v prihodnosti še pomembnejša.

Zahvala

Ker je avtorska zasedba tega poglavja mednarodna, smo rokopis pripravili v angleščini. V slovenščino ga je prevedla Dafne Marko, ki se ji za natančen in tekoč prevod ter skrbno upravljanje s terminologijo iskreno zahvaljujemo.

2 https://www.github.com/clarinsi/

Literatura

Brown, Peter F., Vincent J. Della Pietra,. Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer, 1992: Class-based n-gram models of natural language. *Computational Linguistics* 18/4. 467–479.

Crystal, David, 2011: *Internet linguistics. A student guide*. New York: Routledge.

Čibej, Jaka, Špela Arhar Holdt, Tomaž Erjavec in Darja Fišer, 2018: Ročno označeni korpusi Janes za učenje jezikovnotehnoloških orodij in jezikoslovne raziskave. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 44–73.

Eisenstein, Jacob, 2013: What to do about bad language on the Internet. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. 359–369. http://www.cc.gatech.edu/~jeisenst/papers/naacl2013-badlanguage.pdf

Erjavec, Tomaž, Nikola Ljubešić in Nataša Logar, 2015: The slWaC corpus of the Slovene Web. *Informatica* 39/1. 35.

Erjavec, Tomaž, 2015: *Reference corpus of historical Slovene goo300k 1.2*. Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1025

Erjavec, Tomaž, Nikola Ljubešić in Darja Fišer, 2017: Adapting a state-of-the--art tagger for South Slavic languages to non-standard text. *Proceedings of the EACL workshop*. The 6th Workshop on Balto-Slavic Natural Language Processing, April 4, 2017 Valencia, Spain. Stroudsburg: The Association for Computational Linguistics. 60–68. http://bsnlp-2017.cs.helsinki.fi/bsnlp2017-book.pdf

Erjavec, Tomaž, Nikola Ljubešić in Darja Fišer, 2018: Korpus slovenskih spletnih uporabniških vsebin Janes. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 16–43.

Fišer, Darja, Tomaž Erjavec, Nikola Ljubešić, and Maja Miličević (2015): Comparing the nonstandard language of Slovene, Croatian and Serbian tweets. Smolej, Mojca (ur.): *Obdobja 34. Slovnica in slovar - aktualni jezikovni opis (1. del)*. Ljubljana: Znanstvena založba filozofske fakultete. 225–231.

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan in Noah A. Smith, 2011: Part-of-speech tagging for Twitter: Annotation, features, and experiments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*. Volume 2. pages Association for Computational Linguistics. 42–47.

Heafield, Kenneth, 2011: KenLM: Faster and smaller language model queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Hinrichs, Erhard, Marie Hinrichs, Thomas Zastrow, 2010: WebLicht: Web-Based LRT Services for German. *Proceedings of the Systems Demonstrations at the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*. Uppsala. 25–29.

Krek, Simon in Erjavec, Tomaž, 2009: Standardised Encoding of Morphological lexica for Slavic languages. *MONDILEX Second Open Workshop*. Kyiv, Ukraine. 24–29

Krek, Simon, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek in Nanika Holz, 2015: *Training corpus ssj500k 1.4*. Slovenian language resource repository CLARIN.SI.

Logar Berginc, Nataša, Miha Grčar, Marko Brakus, Tomaž Erjavec, Špela Arhar Holdt in Simon Krek, 2012: *Korpusi slovenskega jezika Gigafida, KRES, ccGigafida in ccKRES: gradnja, vsebina, uporaba*. Ljubljana: Trojina, zavod za uporabno slovenistiko: Fakulteta za družbene vede.

Ljubešić, Nikola, Marija Stupar, Tereza Jurić, in Željko Agić, 2013: Combining available datasets for building named entity recognition models of Croatian and Slovene. *Slovenščina 2.0* 1/2. 35–57.

Ljubešić, Nikola, Tomaž Erjavec in Darja Fišer, 2014: Standardizing Tweets with Character-Level Machine Translation. Gelbukh, Alexander (ur.): *CICLing, Lecture notes in computer science*. Berlin, Heidelberg: Springer. 164–175.

Ljubešić, Nikola, Tomaž Erjavec in Darja Fišer, 2018: Orodja za procesiranje nestandardne slovenščine. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 74–99.

Ljubešić, Nikola, Darja Fišer, Tomaž Erjavec, Jaka Čibej, Dafne Marko, Senja Pollak in Iza Škrjanec, 2015: Predicting the Level of Text Standardness in User-generated Content. *Proceedings of Recent Advances in Natural Language Processing*. 371–378.

Ljubešić, Nikola, Tomaž Erjavec in Darja Fišer, 2016: Corpus-based diacritic restoration for south slavic languages. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA). 3612–3616.

Ljubešić, Nikola in Tomaž Erjavec, 2016: Corpus vs. lexicon supervision in morphosyntactic tagging: the case of Slovene. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA). 1527–1531.

Ljubešić, Nikola, Filip Klubička, Željko Agić, Ivo-Pavao Jazbec, 2016a: New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).*

Ljubešić, Nikola, Tomaž Erjavec, Darja Fišer, Erhard Hinrichs, Marie Hinrichs, Cyprian Laskowski, Filip Petkovski in Wei Qui, 2017: Multilingual Text Annotation of Slovenian, Croatian and Serbian with WebLicht. *Proceedings of the CLARIN Annual Conference*. 18–20 September, Budapest, Hungary.

Martinc, Matej, Senja Pollak in Ana Zwitter Vitez, 2018: Delotoki za nadaljnje analize nestandardne slovenščine. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 16–43.

Plank, Barbara, Anders Sogaard in Yoav Goldberg, 2016: Multilingual Part-of--Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, August 7-12, 2016. 412–418.

Štajner, Tadej, Tomaž Erjavec in Simon Krek, 2013: Razpoznavanje imenskih entitet v slovenskem besedilu. *Slovenščina 2.0* 1/2. 58–81.

Tjong Kim Sang, Erik, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, Feike Dietz, Stefanie Dipper, Miguel Domingo, Rob van der Goot, Marjo van Koppen, Nikola Ljubešić, Robert Östling, Florian Petran, Eva Pettersson, Yves Scherrer, Marijn Schraagen, Leen Sevens, Jörg Tiedemann, Tom Vanallemeersch in Kalliopi Zervanou, 2017: The CLIN27 Shared Task: Translating Historical Text to Contemporary Language for Improving Automatic Linguistic Annotation. *Computational Linguistics in the Netherlands Journal* 7/1. 53–64.

Scherrer, Yves in Nikola Ljubešić, 2016: Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. *Proceedings of the 13th Conference on Natural Language Processing (KONVENS).* September 19-21, 2016, Bochum, Germany. 248–255. https://www.linguistics.rub.de/konvens16/pub/32_konvensproc.pdf