Pomembnost realistične evalvacije: primer popravkov sklona in števila v slovenščini z velikim jezikovnim modelom

Timotej PETRIČ
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Špela ARHAR HOLDT
Filozofska fakulteta in Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Marko ROBNIK-ŠIKONJA
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Med napake pri pisanju v standardni slovenščini sodi raba neustreznega slovničnega sklona ali števila. S pomočjo velikega jezikovnega modela SloBERTa smo razvili novo metodologijo za strojno prepoznavo tovrstnih težav, ki smo jo preizkusili na neustrezni rabi tožilnika namesto rodilnika in množine namesto dvojine. Za vrednotenje in spreminjanje besednih oblik v vhodnih povedih smo uporabili standardna orodja za obdelavo naravnega jezika, kot sta oblikoskladenjski označevalnik CLASSLA-Stanza in leksikon besednih oblik Sloleks. Predlagani popravki temeljijo na statistiki besednih oblik pri uporabi napovedovanja maskirane besede z velikim jezikovnim modelom. Zaradi pomanjkanja zadostne količine učnih podatkov smo napovedne modele učili na umetno generiranih napakah. Uspešnost strojnega popravljanja smo najprej ovrednotili na umetnih množicah in korpusu Lektor, kasneje pa še na novoustvarjeni evalvacijski množici Šolar-Eval. Evalvacija na prvih dveh množicah

Petrič, T; Arhar Holdt, Š.; Robnik-Šikonja, M.: Pomembnost realistične evalvacije: primer popravkov sklona in števila v slovenščini z velikim jezikovnim modelom. Slovenščina 2.0, 12(1): 106–130.

1.01 Izvirni znanstveni članek / Original Scientific Article

DOI: https://doi.org/10.4312/slo2.0.2024.1.106-130

je pokazala visoko uspešnost razvite metodologije (zaznanih več kot 90 % napačno nastavljenih besed), Šolar-Eval pa je razkril mnogo slabšo uspešnost na realističnih podatkih (zaznanih le 29,5 % težav tipa rodilnik-tožilnik in 11,4 % težav tipa dvojina-množina). V celoti rezultati kažejo na nevarnost pretiranega prilagajanja podatkovnim množicam in pomembnost evalvacije na ciljno grajenih avtentičnih podatkih, ki pa so za slovenščino še vedno pomanjkljivi.

Ključne besede: strojno slovnično pregledovanje, slovnični sklon, slovnično število, veliki jezikovni modeli, evalvacija

1 Uvod

Slovnični pregledovalniki – programi, ki preverjajo slovnično ustreznost pisnih besedil, opozarjajo na potencialne jezikovne težave in predlagajo popravke – so ena od temeljnih jezikovnih tehnologij in predstavljajo pomembno digitalno infrastrukturo za sodobne jezike, tudi slovenščino (Krek, 2023). Tipična težava, na katero slovnični pregledovalniki opozarjajo, je raba besednih oblik, ki glede na kontekst ne ustrezajo po sklonu, številu ali kaki drugi slovnični lastnosti. Za preizkus novega pristopa k strojnemu slovničnemu pregledovanju smo izbrali neustrezno rabo tožilnika namesto rodilnika ter množine namesto dvojine. Tovrstne menjave oblik so v procesu razvoja jezikovnih kompetenc v standardni slovenščini relativno pogoste; frekvenčni seznam jezikovnih popravkov v razvojnem korpusu Šolar 3.0 (Arhar Holdt idr., 2022b) pokaže, da so popravki menjave rodilnika in tožilnika na 2. mestu vseh oblikoslovnih kategoričnih popravkov (predstavljajo 341 od 2.916 popravkov), popravki dvojine in množine pa na 5. mestu (246 od 2.916 popravkov), pri čemer podatki kažejo tudi trdoživost obravnavanih jezikovnih problemov, ki se pojavljata tako v osnovni kot vse do konca srednje šole.[1]

V zadnjih letih je povečana zmogljivost vzporednega računanja z grafičnimi procesorji povzročila nov val uspehov na področju umetne inteligence, tudi pri obdelavi naravnega jezika. Trenutno najpomembnejša arhitektura nevronskih mrež, ki prevladujejo pri obdelavi jezika,

1 Na ostalih prvih mestih so heterogenejše skupine jezikovnih težav: na 1. mestu različne menjave sklonov, ki v označevalnem sistemu niso dobile lastne označevalne kategorije, na 3. mestu raznoliki primeri menjav ednine in množine, na 4. mestu pa raznoliki popravki glagolskega časa.

je *transformer* (Vaswani idr., 2017). Modeli, kot je BERT (Devlin idr., 2019), se z napovedovanjem manjkajočih besed v povedih na veliki količini gradiva naučijo jezikovnih značilnosti besedil. V članku smo tak slovenski model, imenovan SloBERTa (Ulčar in Robnik-Šikonja, 2021b), uporabili za generiranje predlogov popravkov neustrezno rabljenega slovničnega sklona in števila.

Ideja predlaganega pristopa poskusi izkoristiti zmožnost modela SloBERTa, da napoveduje maskirane (skrite) besede v povedi. Pristop najprej poišče potencialno problematične besede, ki jih želimo strojno preveriti, v našem primeru besede v tožilniku ali množini. Te besede obravnavamo kot skrite. S pomočjo modela SloBERTa napovemo možne besede na tem mestu povedi in s pomočjo označevalnika CLASSLA-Stanza (Ljubešić in Dobrovoljc, 2019) pridobimo njihove oblikoskladenjske lastnosti. Iz statistike napovedanih oblikoskladenjskih lastnosti nato napovemo najverjetnejši potencialni popravek oblike izvorne besede, npr. če je večina napovedanih besed v dvojini, potencialno problematična beseda pa je v množini, predlagamo njen popravek v dvojino. Besedno obliko z želenimi oblikoskladenjskimi lastnostmi, ki jo program predlaga kot ustrezno, pridobimo iz oblikoslovnega leksikona Sloleks (Dobrovoljc idr., 2019).

Pristop je metodološko nov, preliminarna evalvacija na umetnih podatkih je pokazala, da je uspešen in potencialno uporaben za različne vrste oblikoslovnih napak. Najprej smo ga ovrednotili na povedih iz korpusa Lektor (Popič, 2014), ko se je pojavila nova evalvacijska množica Šolar-Eval (Gantar idr., 2023), pa še na njej. Evalvacijo na korpusu Lektor smo izvedli tako, da smo v povedih nastavili različno število besed v (obravnavanem) napačnem sklonu ali številu. Izračunane metrike natančnosti, priklica in ocene F_1 so pokazale dobro pravilnost in potencialno praktično uporabnost predlaganih popravkov. Evalvacijo na korpusu Šolar-Eval smo izvedli kvalitativno, z identifikacijo in analizo zaznanih in nezaznanih napak. Ti rezultati so, v nasprotju s prejšnjimi, pokazali dokaj nizko uspešnost preizkušenega pristopa. Kljub temu menimo, da raziskava prinaša koristen uvid v pomemben jezikovnotehnološki problem, ki vključuje tudi pomen kakovostne evalvacije in ciljno grajenih evalvacijskih množic.

Članek je sestavljen iz sedmih razdelkov. V razdelku 2 predstavimo

sorodna dela, v razdelku 3 uporabljene jezikovne vire in tehnologije ter v razdelku 4 predlagano metodologijo napovedovanja slovničnih popravkov. V razdelku 5 opišemo postopek evalvacije, v razdelku 6 pa njene rezultate. Članek zaključimo z razdelkom 7, kjer povzamemo opravljeno delo in začrtamo smer za nadaljnje izboljšave.

2 Sorodna dela

Trenutno za slovenski jezik ne obstaja brezplačen slovnični pregledovalnik. Najbolj dodelano orodje, na voljo v uporabniško prijaznem vmesniku, ki napake in popravke tudi vizualizira, je komercialno razvita Amebis Besana.[2] Program, ki temelji na ročno sestavljenih jezikovnih pravilih in podatkovni zbirki Ases (Romih in Holozan, 2002), v brezplačni testni različici omogoča strojno preverbo krajših besedil (do 500 znakov). Vmesnik Besane je podoben komercialnim izdelkom za tuje jezike, kot sta Grammarly in ProWritingAid, ki zaznavajo in popravljajo težave pri rabi ločil, črkovanju in slovnici, prav tako pa ponujajo predloge za izboljšanje sloga pisanja, besedne raznolikosti ter jasnosti in učinkovitosti sporočil.

S strojnimi popravki jezikovnih napak z uporabo globokih nevronskih mrež se je ukvarjalo že več avtorjev. Božič je v okviru diplomskega dela razvil model za avtomatsko popravljanje vejic v slovenskem jeziku (Božič, 2020), ki je pravilno napovedal 92,5 % primerov. Njegov še nekoliko izboljšan program Vejice je prosto dostopen na spletni strani Centra za jezikovne vire in tehnologije Univerze v Ljubljani.[3] Rizvič se je ukvarjal z avtomatskim postavljanjem ločil v tekstu, pridobljenim iz prepoznavalnika govora (Rizvič, 2020). Najboljše rezultate je dosegel z uporabo vektorskih vložitev ELMo in modela BERT. Dosežena ocena F_1 je bila 91,0 %, 91,6 % in 72,0 % za napovedovanje mesta vejice, pike in vprašaja. S predlogi popravkov končnih ločil se je ukvarjal Velikonja (Velikonja, 2021), ki je s pomočjo modela SloBERTa napovedoval tip in mesto postavitve končnih ločil. Naučeni model je dosegel oceno F_1 96,4 % za postavljanje pike in 85,1 % za postavljanje vprašaja. Napovedovanje klicaja ni bilo uspešno.

2 Spletna stran programa: https://besana.amebis.si/.
3 Dostopno na https://orodja.cjvt.si/vejice/.

S pomočjo modelov SloBERTa in SloT5 (Ulčar in Robnik-Šikonja, 2023) ter uporabo orodja CLASSLA-Stanza (Ljubešić in Dobrovoljc, 2019) in leksikona Sloleks (Dobrovoljc idr., 2019) je Mokotar razvil metodologijo za zaznavanje, prepoznavanje in popravljanje različnih vrst jezikovnih napak (Mokotar, 2023), pri čemer je bila uporabljena nekoliko poenostavljena tipologija napak iz korpusa šolskih besedil Šolar (Arhar Holdt idr., 2022a). Modela zaznavanja in prepoznavanja sta dosegla oceni F_1 88 % in 14 %, model za popravljanje pa oceno GLEU 50 %.

Z napovedovanjem in popravljanjem jezikovnih napak v drugih jezikih se je ukvarjalo več avtorjev. Rozovskaya idr. (2014) so razvili sistem za prepoznavanje napačne oblike glagola za angleščino s klasičnimi pristopi strojnega učenja. V zadnjem času se za detekcijo in korekcijo napak uporabljajo izključno nevronski pristopi, predvsem temelječi na velikih jezikovnih modelih. Zhang idr. (2020) so prilagodili arhitekturo modela BERT za napovedovanje jezikovnih napak na primeru kitajščine. Pred kratkim so pregledni članek o strojnem pregledovanju jezikovnih napak pripravili Bryant idr. (2023).

V zadnjem času so se pojavili tudi pristopi s še večjimi velikimi jezikovnimi modeli, kot sta GPT-3 (Brown idr., 2020) in LLaMA (Touvron idr., 2023). V javnosti znani izdelki, kot je ChatGPT, ki v času priprave prispevka uporablja GPT-3.5, so naučeni predvsem na angleškem jeziku. Za slovenščino še ne obstajajo dovolj obsežni jezikovni viri, s katerimi bi naučili primerljivo zmogljiv jezikovni model, zato je uporaba za popravljanje slovenskih besedil trenutno omejena. Evalvacije (Fang idr., 2023; Wu idr., 2023) so pokazale, da je ChatGPT nagnjen k pretiranemu popravljanju oz. spreminjanju izvornega besedila (ang. *over-correcting*), kar je lahko moteče, kadar želimo minimalno in transparentno jezikovno intervencijo (npr. za potrebe jezikovne didaktike).

V tem članku preizkušeni pristop se od zgoraj omenjenih del razlikuje po novi metodologiji, ki uporablja statistiko predlaganih besed maskirnega jezikovnega modela.

3 Uporabljeni viri in tehnologije

Predlagana metodologija za strojno slovnično pregledovanje temelji na jezikovnih virih in tehnologijah, ki jih opisujemo v tem razdelku. V

razdelku 3.1 opišemo jezikovni model BERT, v razdelku 3.2 pa njegovo slovensko različico SloBERTa, ki jo uporabljamo za napovedovanje potencialnih zamenjav maskiranih besed. V razdelku 3.3 predstavimo zbirko orodij CLASSLA-Stanza, ki jo uporabljamo za segmentacijo na povedi in besede ter za oblikoskladenjsko označevanje. Na koncu, v razdelku 3.4, predstavimo še oblikoslovni leksikon Sloleks.

3.1 Model BERT

BERT (Devlin idr., 2019) je vnaprej naučeni nevronski maskirni jezikovni model, ki temelji na arhitekturi *transformer* (Vaswani idr., 2017). Kot odprtokodno ogrodje je na voljo za različne naloge strojne obdelave naravnega jezika. Naučen je s pomočjo velikih besedilnih korpusov in zaradi svoje arhitekture in načina maskiranega učenja vsebuje predstavitev besed v kontekstu. To znanje lahko uporabimo za reševanje mnogih nalog, med drugim strojno označevanje sentimenta, odgovarjanje na vprašanja in tudi napovedovanje manjkajočih besed vhodnega besedila, kar smo uporabili v našem delu.

3.2 Model SloBERTa

Veliki maskirni jezikovni model SloBERTa (Ulčar in Robnik-Šikonja, 2021a; 2021b) uporablja arhitekturo robustne inačice modela BERT, imenovane RoBERTa (*A Robustly Optimized BERT Pretraining Approach*) (Liu idr., 2019). Arhitektura RoBERTa pri učenju namesto statičnega maskiranja besed uporablja dinamično maskiranje, kar pomeni, da se maskiranje besed ne zgodi samo enkrat – v fazi predpriprave vhodnih besedil – ampak večkrat, med posameznimi epohami učenja; RoBERTa tudi opusti nalogo napovedovanja, ali sta dve vhodni povedi sosednji v besedilu, ki je prisotna v modelu BERT. SloBERTa je enojezikovni model, naučen na 3,47 milijarde pojavnic (besed in ločil) iz vhodnih besedil. Slovar pojavnic, ki jih model uporablja za pretvorbo besedila v sezname vektorskih vložitev, ima 32.000 vnosov. Celotno učenje na besedilih izbranih slovenskih korpusov, kot so Gigafida 2.0 (Krek idr., 2020), siParl 2.0 (Pančur idr., 2020) in KAS (Erjavec idr., 2019), je obsegalo 98 epoh. Implementacija modela SloBERTa je med drugim na voljo v programski knjižnici HuggingFace, ki omogoča

odprtodostopen prenos modela ter preprosto uporabo v programskem jeziku Python.[4]

3.3 Zbirka orodij CLASSLA-Stanza

CLASSLA-Stanza (Ljubešić in Dobrovoljc, 2019) je zbirka orodij za procesiranje in jezikoslovno označevanje besedil. Med drugim omogoča segmentacijo, lematizacijo, oblikoskladenjsko in skladenjsko označevanje ter označevanje imenskih entitet v (standardnih in nestandardnih) slovenskih, hrvaških, srbskih, bolgarskih in deloma tudi makedonskih besedilih. Temelji na knjižnici Stanza (Qi idr., 2020). Označevalnik CLASSLA-Stanza v našem delu uporabljamo za segmentacijo besedila na povedi in besede ter oblikoskladenjsko označevanje besednih oblik po sistemu Multext-East v6 (Erjavec, 2017).[5] Sistem vsebuje nabor oznak (na kratko *oznake MSD*), ki določijo besedno vrsto, nato pa niz pri tej besedni vrsti izkazanih slovničnih lastnosti, kot so denimo spol, sklon in število.

3.4 Leksikon Sloleks

Sloleks je odprtodostopni leksikon besednih oblik za slovenščino, ki poleg osnovne oblike besede vsebuje nabor pregibnih oblik, podatke o pogostosti leme in pregibnih oblik iz referenčnega pisnega korpusa, zbir standardnih in nestandardnih oblikoslovnih variant ter povezave na besedotvorno sorodne besede (Dobrovoljc idr., 2015). Verzija 2.0,[6] ki je dostopna na repozitoriju CLARIN.SI (Dobrovoljc idr., 2019), vsebuje 100.805 leksikonskih enot. Oblikoslovni leksikon v našem delu uporabljamo za pridobivanje besednih oblik v želenem sklonu ali številu.

Uporabljena podatkovna množica ima oblikoskladenjske informacije po sistemu MULTEXT-East (Erjavec, 2017). V stolpcih po vrsti vsebuje besedne oblike, leme, oznake MSD in frekvence pojavitve. V zadnjem stolpcu so kategorije, ki jih vsebujejo oznake MSD, izpisane z besedo. Za lažjo predstavo na Sliki 1 prikažemo izsek dela leksikona iz tekstovne datoteke.

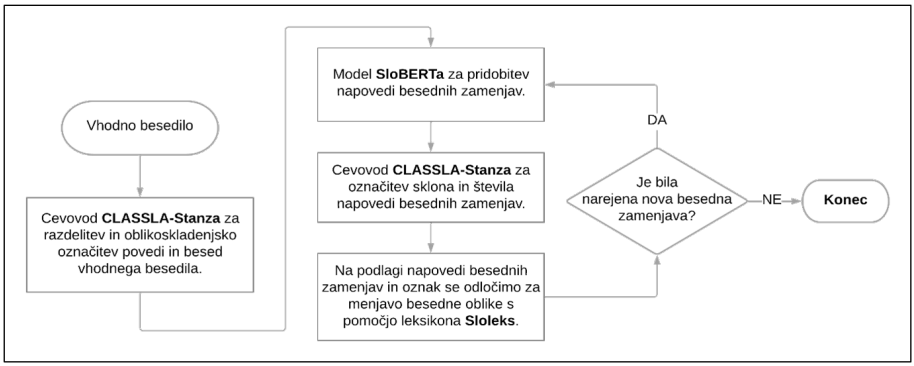
4 Dostopno na https://huggingface.co/EMBEDDIA/sloberta.
5 Verzija 6 je dostopna na http://nl.ijs.si/ME/V6/msd/html/msd-sl.html.
6 V času med razvojem metodologije in objavo prispevka je bila na CLARIN.SI objavljena tudi že nadgrajena različica Sloleks 3.0.

BESEDA	LEMA	MSD	FREKVENCA	PODROBNEJŠI OPIS MSD
abscesa	absces	Ncmsg	0.000011	Noun Type=common Gender=masculine Number=singular Case=genitive NOUN Case=Gen Gender=Masc Number=Sing
absces	absces	Ncmsn	0.000063	...
abscesi	absces	Ncmpn	0.000031	...

Slika 1: Primer informacij v leksikonu Sloleks (vir: Dobrovoljc idr., 2019). Izmed podatkov uporabljamo predvsem stolpce BESEDA, LEMA in MSD. Glede na lemo in želeno spremembo sklona in/ali števila s pomočjo oznake MSD poiščemo ustrezno obliko obravnavane besede.

4 METODOLOGIJA SLOVNIČNEGA POPRAVLJANJA

V tem poglavju predstavimo cevovod za napovedovanje slovničnih popravkov in opišemo posamezne gradnike. Shema cevovoda je prikazana na Sliki 2.



Slika 2: Shema cevovoda za napovedi jezikovnih popravkov. Model SloBERTa napove 10 najverjetnejših besed namesto zamaskirane. To so lahko različne besede (ne le dejansko zamaskirana), vendar za vse, na podlagi konteksta, s pristopom CLASSLA-Stanza napovemo sklon in število. Na podlagi teh napovedi določimo najverjetnejši sklon in število, potem pa za ta sklon in število v Sloleksu poiščemo pravo obliko za zamaskirano besedo.

4.1 Segmentacija in izbor kandidatov za popravke

Vhodno besedilo najprej obdelamo z orodji CLASSLA-Stanza za segmentacijo na povedi in besede. Za vse besede vsake povedi določimo še njihove oblikoskladenjske oznake. Z modelom SloBERTa (ali

SloBERTaMCD, opisanim v razdelku 4.2) za vsako besedo vsake povedi, ki ji je mogoče določiti sklon ali število, preverimo, če je možno pripisati tudi drugačen sklon ali število glede na vrednosti parametrov *caseXY* in *numberXY*, npr. če je parameter *case42* nastavljen na vrednost 1, zamenjave besed iz tožilnika v rodilnik ne iščemo, za vrednosti med 0 in 1 pa jih poskusimo najti; podobno velja npr. za parameter *number32*, kjer vrednost 1 pomeni, da zamenjav iz množine v dvojino ne iščemo, za vrednosti manjše od ena pa sprožimo iskanje potencialnih popravkov. Parameter *caseXY* določa verjetnostni prag prepričanosti modela v trenutni sklon besede.

Za besedo v tožilniku in mejno vrednost parametra nastavljeno na 0,5, torej *case42=0,5* bi denimo iskali njeno zamenjavo v rodilniku in jo uveljavili, če bi prepričanost napovednega modela za to spremembo presegla prag 0,5. Če bi za vse mejne vrednosti izvornega sklona rodilnik veljalo *case4{1,2,3,5,6}=1*, zamenjav za besede v tožilniku ne bi iskali. Na ta način iščemo le zamenjave besed v sklonu ali številu, za katere smo naučili napovedne modele. Trenutno iščemo popravke le za *case42* in *number32* – torej popravke sklona iz tožilnika v rodilnik in števila iz množine v dvojino.

Mejne vrednosti smo naučili v procesu optimizacije hiperparametrov. Omejili smo se le na iskanje optimalnih mejnih vrednosti za *case42, number32* in še za nastavitev kombinacije obeh mejnih vrednosti hkrati: *case42* in *number32*.

4.2 Zanesljivejše napovedi s SloBERTaMCD

Namesto modela SloBERTa lahko za napovedi zamenjav maskiranih besed iz vhodnih besedil uporabimo postopek SloBERTaMCD. Postopek uporablja model SloBERTa (razdelek 3.2), ki ga uporabimo na način Monte Carlo Dropout (MCD) (Miok idr., 2022). Pristop uporablja izpust nevronov (angl. *dropout*) v fazi napovedovanja kot tehniko *regularizacije* (Gal in Ghahramani, 2016). To storimo tako, da vsako poved z maskirano besedo 50-krat obdelamo z modelom SloBERTa in na koncu vrnemo vektor povprečij vseh napovedi besednih zamenjav. Pri uporabi MCD ima izhodni vektor napovedi zamenjav bolje kalibrirane verjetnosti napovedi uteži, kar lahko pomaga pri razvrstitvi in izboru najbolj primernih zamenjav.

4.3 Možne zamenjave

Če za besedo iščemo zamenjave, jo zamaskiramo in celotno poved v obliki niza dodamo v nov seznam povedi (ki imajo vse po eno maskirano besedo). Ta seznam povedi obdelamo z modelom SloBERTa (ali SloBERTaMCD). Kot izhod za vsako vhodno poved s po eno maskirano besedo prejmemo po 10 napovedi najbolj primernih besednih zamenjav (to so lahko poljubne besede). Vsaka napoved ima pripisano verjetnost in besedno zamenjavo.

Za vsako napoved besedne zamenjave pridobimo podatke o sklonu in številu s pomočjo orodja CLASSLA-Stanza. Tokrat je povedi 10-krat več kot pri začetnem označevanju, saj imamo za vsako poved s po eno maskirano besedo 10 možnih besednih zamenjav. CLASSLA-Stanza nam oblikoskladenjsko označi le primere, ki so na mestu izvorno maskirane besede. Za vsako besedo, ki ji je možno pripisati sklon ali število, imamo zdaj po 10 predlogov besednih zamenjav in njihove oblikoskladenjske lastnosti. Kot končni predlog zamenjave izberemo sklon in število, ki imata največjo vsoto verjetnosti. Če imata predlagana sklon in število dovolj veliko vsoto verjetnosti v primerjavi z mejnimi vrednostmi, bomo za izhodiščno besedno obliko predlagali zamenjavo.

4.4 Ponavljanje in ustavitev postopka

Zgoraj opisani postopek ponavljamo, dokler sistem še predlaga nove besedne zamenjave. V vsakem dodatnem obhodu obdelujemo le besede, za katere še nimamo besednih zamenjav. Ponavljanje je potrebno, ker v enem obhodu ponavadi ne naslovimo vseh potencialnih težav, npr. popravimo samostalnik, pridevnika pred njim pa ne. Primer zaznanih napak (v rdečem), predlaganih popravkov (v zelenem) in izpisa po koncu procesiranja prikazujemo spodaj. Kot vidimo v zadnjih dveh povedih, se lahko zgodi tudi, da sistem ne zazna vseh napak v besedilu.

Hotel je preizkusiti dve testne|testni vožnje|vožnji z avtom. Opazil je, da mu manjkata dve iztočnice|iztočnici. Včeraj nisem videl Petro|Petre. Že dolgo nisem jedel tako dobro|dobre solato|solate. Ne morem jo videti! Naredil je dve čudne|čudni napake.

Hotel je preizkusiti dve **testni vožnji** z avtom. Opazil je, da mu manjkata **dve iztočnici.** Včeraj nisem videl **Petre.** Že dolgo nisem jedel tako **dobre solate.** Ne morem **jo** videti! Naredil je dve **čudni napake.**

5 Postopek evalvacije

V razdelku 5.1 opišemo korpus Lektor, ki ga uporabljamo za evalvacijo umetno ustvarjenih podatkov in učenje mejnih vrednosti parametrov. Šolar-Eval, ki ga uporabljamo za kvalitativno evalvacijo na avtentičnem gradivu, opišemo v razdelku 5.2. Postopek nastavljanja napačnih besed za evalvacijo opišemo v razdelku 5.3. Razdelek 5.4 predstavi uporabljene evalvacijske metrike.

5.1 Korpus Lektor

Lektor (Popič, 2014) je korpus lektoriranih avtorskih besedil in prevodov. V njem so zbrana novejša neliterarna (strokovna in poljudnoznanstvena) besedila različnih avtorjev in prevajalcev, ki so jih lektorirali različni lektorji. Vsako korpusno besedilo vsebuje metapodatke o avtorju, publikaciji, lektorju, pripisane pa ima tudi leme, oblikoskladenjske oznake in vsebinske kategorije lektorskih popravkov. Kot podatkovno bazo v formatu XML smo ga za raziskovalne namene dobili od avtorja korpusa Damjana Popiča.

Korpus smo pretvorili v tekstovno datoteko, kjer je v vsaki vrstici po ena lektorirana poved – skupno 28.744 povedi. Povedi so po vrsticah razvrščene naključno. Datoteko smo razdelili na dva dela: prvi del ima 80 % oz. 22.995 vseh povedi in je bil uporabljen za učenje mejnih vrednosti parametrov, drugi ima preostalih 20 % oz. 5.749 povedi in je bil uporabljen pri evalvaciji programske rešitve.

5.2 Korpus Šolar-Eval

Med delom se je izkazalo, da obstoječi korpusi z jezikovnimi popravki ne vsebujejo dovolj podrobno ali dosledno označenih podatkov, da bi bili neposredno uporabni pri razvoju slovenskih črkovalnikov in slovničnih pregledovalnikov (Gantar idr., 2023: 91). Kot rešitev je bila pripravljena evalvacijska množica Šolar-Eval (Arhar Holdt idr., 2023). Ta

vsebuje 109 esejev, ki so jih napisali slovenski osnovnošolci in srednješolci, ter vključuje 9.808 jezikovnih popravkov, ki so jih podrobno in usklajeno označili jezikoslovci. Vsebuje tudi metapodatke o korpusnih besedilih in raznolike jezikoslovne oznake.

V evalvacijo smo vključili povedi, v katerih je najti avtentične popravke tožilnika v rodilnik (46 povedi), množine v dvojino (57 povedi) in oboje (1 poved). Številne od teh povedi vsebujejo tudi težave in popravke drugih jezikovnih značilnosti.

5.3 Nastavljanje napačnih besednih oblik za evalvacijo

Za model smo naučili tri kombinacije mejnih vrednosti parametrov: mejno vrednost *case42, number32* in sočasno nastavitev kombinacije obeh mejnih vrednosti *case42* in *number32* (opisano v razdelku 4.1). Vsako od kombinacij mejnih vrednosti parametrov smo evalvirali posebej. Za evalvacijo smo izbrali 3.030 povedi izmed 5.749 iz testne množice korpusa Lektor (opisana v razdelku 5.1). V množici je bilo skupno 61.180 besed, med njimi 7.629 besed v tožilniku, 9.060 v rodilniku, 12.732 v množini in 690 v dvojini.

Za vsako besedo, ki je bila v iskanem sklonu ali številu, smo poiskali vse možne oblike te besede v neustreznem sklonu oz. številu in jih shranili v seznam. Na primer, ko smo evalvirali model z nastavljeno mejno vrednostjo *case42*, smo za vse besede v rodilniku poiskali alternativne besedne oblike v tožilniku. Podobno smo storili tudi za evalviranje modela z nastavljeno mejno vrednostjo *number32*. Ko smo evalvirali model s hkratno nastavitvijo obeh mejnih vrednosti, smo za vsako besedo v seznam shranili vse možne alternativne besedne oblike (tožilnik ali množina). Shranjevali smo samo alternativne besedne oblike, ki so se razlikovale od izvornih (ne pa tudi enakopisnih oblik).

Napačne besedne oblike smo nastavili, ko je bilo možno za izbrano poved pridobiti vsaj toliko besed z alternativnimi besednimi oblikami, kot smo želeli. Če je bilo za določeno besedo na voljo več alternativnih oblik, smo naključno izbrali le eno. Nastaviti napačno besedno obliko torej pomeni, da smo za ustrezno obliko v rodilniku nastavili še neustrezno obliko v tožilniku. Kasneje je program za vse

primere, ki so bili v tožilniku že izvorno ali pa so imeli nastavljene tožilniške oblike, iskal popravke iz tožilnika v rodilnik (oz. iz množine v dvojino).

5.4 Ocenjevanje modela

Najprej smo programsko rešitev testirali s pomočjo podatkov, ustvarjenih iz korpusa Lektor. Za evalvacijo napovednih modelov smo v povedih nastavili različno število besed v želenem napačnem sklonu ali številu. Izbrali smo samo povedi, v katerih se je neustrezna besedna oblika formalno razlikovala od ustrezne (ne pa primerov z enakopisnimi oblikami). Povedi smo obdelali s cevovodom, opisanim v razdelku 4, in dobili strojno predlagane slovnične popravke.

Rezultate smo razdelili v štiri skupine. Besede, ki so imele nastavljeno napačno obliko in dobile ustrezen popravek, smo označili za dejansko pozitivne primere (angl. *true positive oz. TP*). Besede, ki so imele nastavljeno napačno obliko in nepravilen ali nenastavljen popravek, smo označili za napačno negativne primere (angl. *false negative oz. FN*). Besede, ki niso imele nastavljene napačne oblike, vendar so imele popravek, smo označili za napačno pozitivne primere (angl. *false positive oz. FP*). Besede, ki niso imele nastavljene napačne oblike in tudi ne popravka, pa smo označili za dejansko negativne (angl. *true negative oz. TN*).

Iz teh štirih skupin smo izračunali metrike natančnost (angl. *precision*), priklic (angl. *recall*) in oceno F_1 (Jurafsky in Martin, 2024).

V naslednjem primeru povedi je 14 besed. Zeleno so obarvane izvorne besede, rdeče so nastavljene napačne besedne oblike in modre so predlogi popravkov, ki so izhod iz programa. Besedi *znamenite* in *trikotne* sta šteti kot dejansko pozitivni. Beseda *Amerikami* je šteta za napačno negativni primer, saj program ni predlagal nobenega popravka. Ostale besede so štete za dejansko negativne.

Trgovina s sužnji je postala del znamenite|znamenito|znamenite trikotne|trikotno|trikotne trgovine med Amerikama|Amerikami, Evropo in Afriko.

V naslednjem koraku smo postopek evalvirali še na avtentičnem gradivu korpusa Šolar-Eval. Korpusne povedi smo obdelali s

cevovodom, opisanim v razdelku 4, in dobili strojno predlagane slovnične popravke. Obravnavane 104 povedi smo strojno slovnično pregledali še s spletno različico slovničnega pregledovalnika Amebis Besana 4.32.8. Oboje rezultate smo primerjali z jezikoslovnimi popravki, podanimi v korpusu, in ročno analizirali podobnosti in razlike.

Med analizo smo popravke razdelili v dve skupini: (a) primeri, v katerih je jezikovna težava opazna in rešljiva na ravni same povedi, npr. *Potem pa sva se ob skodelici mamine kave pogovarjale o imenih, barvi las in podobno*, (b) primeri, kjer je za popravek treba poznati širši besedilni kontekst ali pa sta v standardnem jeziku sprejemljivi tako popravljena kot nepopravljena različica, npr. *Zgodbe sem prebral ker sem jih moral*, pri čemer je iz predhodnega besedila razvidno, da je učenec prebral samo dve zgodbi. Ker gre za različno zahtevne primere, jih v Rezultatih prikažemo ločeno.

6 Rezultati

V tem razdelku predstavimo rezultate evalvacije. Najprej v razdelku 6.1 predstavimo rezultate na korpusu Lektor z nastavljenimi napakami tipa tožilnik-rodilnik in množina-dvojina. Sledijo rezultati evalvacije na avtentičnem gradivu v razdelku 6.2, nato pa še evalvacija hitrosti obdelave v razdelku 6.3.

6.1 Rezultati evalvacije na umetnem gradivu

V Tabelah 1, 2 in 3 so rezultati testiranja na množici s 3.030 povedmi, kjer smo primerjali tudi uporabo napovedovanja z uporabo metode MCD (oz. SloBERTaMCD iz razdelka 4.2) v primerjavi z običajnim modelom SloBERTa (v tabelah označeno z *MCD* in *neMCD*).

Ker vse povedi iz testne množice nimajo vedno iskanega števila besed v konfiguraciji pravilnega sklona oz. števila (npr. nimamo povedi z vsaj tremi besedami v pravilnem rodilniku, ki jim lahko nastavimo nepravilno obliko v tožilniku), se zgodi, da se za te povedi ne nastavijo nobene napačne besede. Te povedi vseeno vključimo v evalvacijo, ker model za besede v iskanem izvornem sklonu oz. številu še vedno poizkusi najti popravek – torej jih lahko beležimo kot potencialno napačno pozitivne primere. Prav tako teh povedi nismo

želeli odstraniti ali spreminjati zato, ker bi s tem vnašali dodatno pristranskost.

Metrike v Tabeli 1 prikazujejo rezultate popravljanj slovničnega števila. V primerjavi s popravljanjem sklonov (Tabela 2) prikazujejo občutno slabše rezultate. To je verjetno posledica dejstva, da je bilo besed v dvojini, ki bi jih lahko uporabili za nastavljanje napačnih besed in evalvacijo, v testni množici bistveno manj. Prav tako je raba dvojine v slovenščini manj pogosta kot raba tožilnika, zato so imeli tudi besedilni korpusi, ki so bili uporabljeni v procesu vnaprejšnjega učenja modela SloBERTa, na voljo manj tovrstnega gradiva.

Tabela 1: *Natančnost napovedi samo za popravljanje slovničnega števila* ***množina-dvojina*** *z različnim številom nastavljenih napačnih besed v povedi v množini*

	Št. napačnih besed v povedi					
	1		**2**		**3**	
	MCD	**neMCD**	**MCD**	**neMCD**	**MCD**	**neMCD**
Napačne besede	171	171	212	212	180	180
Natančnost	89,3	**90,4**	91,8	**92,2**	90,7	**90,9**
Priklic	**78,4**	77,2	**79,7**	78,3	75,6	**77,8**
F-ocena	**83,5**	83,3	**85,4**	84,7	82,4	**83,8**

**Opomba.* Model je imel nastavljeno mejno vrednostjo number32=0,056 (oz. number32=0,032 za neMCD).

Tabela 2: *Natančnost napovedi samo za popravljanje sklona* ***tožilnik-rodilnik*** *z različnim številom nastavljenih napačnih besed v povedi v tožilniku*

	Št. napačnih besed v povedi					
	1		**2**		**3**	
	MCD	**neMCD**	**MCD**	**neMCD**	**MCD**	**neMCD**
Napačne besede	1.994	1.994	2.858	2.858	2.853	2.853
Natančnost	98,2	**98,4**	98,5	**98,9**	98,4	**98,7**
Priklic	**96,2**	95,4	93,3	**93,5**	**92,6**	92,5
F-ocena	**97,2**	96,9	95,8	**96,1**	95,4	**95,5**

**Opomba.* Model je imel z nastavljeno mejno vrednostjo case42=0,0097 (oz. case42=0,007 za neMCD).

Tabela 3: *Natančnost napovedi za popravljanje sklona **tožilnik-rodilnik** in števila **množina-dvojina***

	Št. napačnih besed v povedi					
	1		2		3	
	MCD	**neMCD**	**MCD**	**neMCD**	**MCD**	**neMCD**
Napačne besede	2.051	2.051	2.986	2.986	3.075	3.075
Natančnost	97,3	**97,7**	98,0	**98,5**	98,0	**98,3**
Priklic	**95,4**	95,0	91,4	**92,6**	91,4	**92,4**
F-ocena	**96,3**	**96,3**	94,6	**95,5**	94,6	**95,3**

**Opomba.* Model je imel nastavljene mejne vrednosti case42=0,006 in number32=0,03 (oz. case42=0,0013 in number32=0,036 za neMCD).

Rezultati v Tabeli 3 kažejo, da se je model dobro naučil napovedovanja besednih zamenjav. V testni množici 3.030 povedi je približno 13-krat več besed v rodilniku kot besed v dvojini. Naučene mejne vrednosti *case42=0,006* in *number32=0,03* (oz. *case42=0,0013* in *number32=0,036* za neMCD) kažejo na to, da se je model v procesu učenja mejnih vrednosti naučil, da besedam v množini postavi višji prag kot besedam v tožilniku. Opozorimo naj, da mejne vrednosti pragov ne predstavljajo kalibriranih verjetnosti, zato jih ne presojamo kot gotovosti modelov za dane odločitve.

Primerjava med napovedovanjem z in brez MCD pokaže dokaj majhne in nekonsistentne razlike. Glede na precej večjo računsko zahtevnost metode MCD zato metode napovedovanja z MCD za naš problem ne moremo priporočiti.

6.2 Rezultati evalvacije na avtentičnem gradivu

Rezultate kvalitativne evalvacije na gradivu korpusa Šolar-Eval prikazujeta Tabeli 4 in 5. Kot je bilo omenjeno v razdelku 5.4, so rezultati ločeni glede na to, ali gre za primere, ki so rešljivi na ravni same povedi, ali primere, kjer je za interpretacijo potrebno več sobesedila.

Tabela 4: *Število avtentičnih jezikovnih popravkov rabe množine namesto dvojine v 58 obravnavanih povedih in število ustreznih strojnih rešitev, ki jih dobimo bodisi samo z novim pristopom, samo z Besano, z obema programoma ali z nobenim od njiju*

	Št. problemov	Novi pristop	Besana	Oba	Nerešeno
Dvoumni primeri	63	**1**	0	0	62
Nedvoumni primeri	25	**7**	3	2	13
Skupaj	88	**8**	3	2	75

Tabela 5: *Število avtentičnih jezikovnih popravkov rabe tožilnika namesto rodilnika v 47 obravnavanih povedih in število ustreznih strojnih rešitev, ki jih dobimo bodisi samo z novim pristopom, samo z Besano, z obema programoma ali z nobenim od njiju*

	Št. problemov	Novi pristop	Besana	Oba	Nerešeno
Dvoumni primeri	7	**1**	0	0	6
Nedvoumni primeri	54	8	**24**	9	13
Skupaj	61	9	**24**	9	19

Tako novi pristop kot Besana pri dvoumnih primerih dosegata izredno nizko uspešnost, vendar tudi nedvoumni primeri z obema postopkoma pogosto ostajajo nezaznani: to velja za 13 od 25 popravkov množine v dvojino ter za 13 od 54 popravkov tožilnika v rodilnik. Pri tem je treba izpostaviti, da na uspešnost popravljanja lahko vplivajo tudi preostali odstopi od standarda, ki se pojavljajo v obravnavanih povedih. Za ponazoritev navajamo nekaj primerov neuspešno identificiranih problemov:

- Ko smo šli prek železnice jaz in prijatel **smo ušli** vlaku, ampak ona pa je bila skoraj pod vlakom.
- Vendar se Hamlet izogiba **vsak kontakt** z Ofelijo.
- Ampak usaj nimaš slabe vesti o tem, kar si naredil in **kar** nisi.

Kot je razvidno iz Tabel 4 in 5, je novi pristop pri problemu dvojine malenkost boljši od Besane, medtem ko je pri težavah rodilnika Besana znatno uspešnejša. Jezikoslovna analiza ni pokazala vzorcev, ki bi nakazovali razloge za (ne)uspešnost enega ali drugega pristopa pri posameznih primerih. Za ponazoritev navajamo tri primere, ki jih je rešil novi pristop, ne pa Besana:

- Ko smo prišli do dveh aprtmajev, ki smo **jih** rezervirali.
- Oba cutita ljubezen drug do drugega ampak jih ta verski spopad ločuje drug od drugega.
- Izhaja iz bolj revne družine, zato si **marsikaj** nemore privoščiti.

Nato pa še tri primere, ki jih je rešila Besana, ne pa novi pristop:

- Če se zaposlim nemorem pričakovati **zlati zaslužek**.
- Ob problemu ne dajo samo **kruto kazen** ampak se o tem pogovorijo in jim svetujejo.
- Jaz in Sternfeldovka sva sklenili, da bo ona šla v grm in ko bo čas bo prišla ven, da bova Tulpenhajna **razkrinkale**.

6.3 Hitrost obdelave besedil

Za nekatere vrste uporabe prepoznavalnikov in popravljalnikov jezikovnih napak je pomembna tudi hitrost izvajanja, npr. pri popravljalnikih, ki delujejo v okviru sprotnega prepoznavanja govora, zato v omejenem obsegu analiziramo tudi ta aspekt razvitih modelov. Hitrost obdelave vhodnih besedil je odvisna predvsem od časa, ki ga program potrebuje, da besedilo (večkrat) obdela z modelom SloBERTa oz. SloBERTaMCD in orodji CLASSLA-Stanza. Meritve smo opravili na računalniku z dvema procesorjema *Intel(R) Xeon(R) Silver 4214*, torej z 48 nitmi. Na računalniku so bile 4 GPE *NVIDIA GeForce RTX 2080 Ti*. Oba modela smo naložili na eno GPE, istočasno pa se na kartici niso izvajali drugi programi.

Ob uporabi SloBERTaMCD smo v povprečju ob obdelavi besedila velikosti približno 61.000 besed dosegli hitrost obdelave 13,06 besed na sekundo. Ob uporabi navadnega modela SloBERTa se je hitrost podvojila na 26,02 besed na sekundo. Hitrost je bila višja, ker SloBERTaMCD uporablja večkratno povprečenje izhodnih vrednosti (opisano v razdelku 4.2) in je napovedovanje besednih zamenjav zato počasnejše.

7 ZAKLJUČEK

Razvili smo metodologijo za napovedovanje slovničnih popravkov, ki temelji na maskirnem jezikovnem modelu. Model vrne verjetnostno distribucijo popravkov slovničnega števila in sklonov, na podlagi katere ocenimo vrsto popravka in njegovo verjetnost. Za dejansko predlagane popravke uporabljamo parameter, katerega vrednost predstavlja prag verjetnosti ustreznosti popravka. Za razvoj metodologije smo uporabili orodje CLASSLA-Stanza (razdelek 3.3), napovedi različnih

besednih oblik z modelom SloBERTa (razdelek 4.2) in slovenski oblikoslovni leksikon Sloleks (razdelek 3.4). Za izboljšanje napovedi verjetnostne distribucije napak smo neuspešno preizkusili metodo MCD, ki je sicer okoli dvakrat počasnejša od napovedi običajnega modela SloBERTa. Izvorna koda programa je odprtodostopna.[7] Metodologijo smo preizkusili na popravkih tipa množina-dvojina in tožilnik-rodilnik in na različnih evalvacijskih množicah dosegli zelo različne rezultate.

Program ob hkratni uporabi naučenih mejnih vrednosti za popravljanje sklona tožilnik-rodilnik in števila množina-dvojina na umetnih podatkih doseže F_1-oceno med 95 % in 96 %. Pravilno je popravil od 92 % do 95 % napačno nastavljenih besed – odvisno od števila nastavljenih napačnih besed v evalvacijski množici. Na drugi strani na avtentičnih podatkih, ki so lahko vpeti v sobesedilo in vsebujejo tudi druge odstope od jezikovnega standarda, program uspešno naslovi le 29,5 % težav z rodilnikom (oz. 31,5 % primerov, ki so rešljivi brez širšega konteksta) in 11,4 % težav z dvojino (oz. 36 % primerov, ki so rešljivi brez širšega konteksta). Če je prva evalvacija kazala na presežno uspešnost predlaganega pristopa, pa je evalvacija na avtentičnem jeziku ugotovitve ustrezno relativizirala.

Rezultati našega dela kažejo, da je evalvacije na umetno pripravljenih podatkih, ki so na področju obdelave naravnega jezika sicer pogoste, nujno dopolnjevati z analizami na avtentičnem jeziku. Kvalitetna evalvacija bi morala upoštevati širši kontekst uporabe modelov, saj osredotočenost na ozko področje določene evalvacijske množice lahko vodi v pretirano prilagajanje specifični jezikovni situaciji na račun širše potencialne uporabe. Raba samo umetnih množic in množic, ki se le delno prekrivajo z nameravanim področjem uporabe, lahko privede do preveč optimističnih ocen uspešnosti delovanja. Na drugi strani je izziv raznolikost izražanja v naravnem jeziku, saj »pravilni odgovori« v jezikovnih podatkih vsebujejo le eno od izraznih možnosti, ne pa drugih prav tako pravilnih možnosti. Ta značilnost lahko modele strojnega učenja vodi do preozke usmerjenosti in zanemari variantne ali kreativnejše možnosti izražanja. Pri evalvaciji lahko vodi do preveč pesimističnih ocen uspešnosti delovanja ali do favoriziranja modelov, ki so preozko usmerjeni.

7 Na voljo na: https://github.com/timopetric/EssayHelper.

Izziv pri evalvaciji nalog s področja obdelave in razumevanja naravnega jezika sta tudi subjektivnost in (ne)konstistentnost človeških odločitev, ki se v evalvacijskih podatkih obravnavajo kot pravilne. Problem je še večji, če metodologija priprave ni transparentno popisana ali množice niso javno na voljo, da bi bile evalvacije ponovljive in razložljive. Ker je priprava kvalitetnih evalvacijskih virov zahtevna in zamudna, jih za številne jezike, tudi slovenščino, še vedno primanjkuje. Šolar-Eval, ki je bil pripravljen posebej za evalvacijo slovenskih črkovalnikov in slovničnih pregledovalnikov, ustreza vsem želenim kriterijem, vendar vsebuje izključno besedila šolajoče se populacije. Podobne množice bi bilo po enotni metodologiji treba v prihodnje razviti tudi za druge vzorce jezikovne rabe.

Ker so se v zadnjem času pojavili mnogo večji generativni jezikovni modeli, kot sta GPT-4 in Gemini, nekateri tudi prostodostopni, npr. LLama-3, se nadaljnji razvoj predstavljene ideje ne zdi smiseln, razen morebiti za samostojno delovanje slovničnih pregledovalnikov na računalnikih z malo računskimi viri. V nadaljnjem delu se bomo osredotočili predvsem na večje odprtodostopne modele in na razvoj učnih množic zanje. Ti modeli lahko, ob ustreznem učenju, naslovijo tudi v članku izpostavljeno in v jeziku pogosto težavo preozkega konteksta, ko je za ustrezno identifikacijo jezikovne težave treba upoštevati širše značilnosti sobesedila, ne le posamezne povedi.

Zahvala

Dostop do celotnega korpusa Lektor nam je omogočil vodja projekta izgradnje korpusa, doc. dr. Damjan Popič, s Filozofske fakultete Univerze v Ljubljani. Raziskovalni program Jezikovni viri in tehnologije za slovenski jezik (P6-0411) in projekta Empirična podlaga za digitalno podprt razvoj pisne jezikovne zmožnosti (J7-3159) in Veliki jezikovni modeli za digitalno humanistiko (GC-0002) sofinancira Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije iz državnega proračuna.

Literatura

Arhar Holdt, Š., Rozman, T., Stritar Kučuk, M., Krek, S., Krapš Vodopivec, I., Stabej, M., Pori, E., ..., & Kosem, I. (2022a). *Developmental corpus Šolar 3.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1589

Arhar Holdt, Š., Rozman, T., Stritar Kučuk, M., Krek, S., Krapš Vodopivec, I., Stabej, M., Pori, E., ..., & Kosem, I. (2022). *Frequency list of language problems from Šolar 3.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1716

Arhar Holdt, Š., Gantar, P., Bon, M., Gapsa, M., Lavrič, P., & Klemen, M. (2023). *Dataset for evaluation of Slovene spell- and grammar-checking tools Šolar-Eval 1.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1902

Božič, M. (2020). *Globoke nevronske mreže za postavljanje vejic v slovenskem jeziku* (Diplomska naloga). Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Pridobljeno s https://repozitorij.uni-lj.si/IzpisGradiva.php?id=119034&lang=slv

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, ..., Askell, A., idr. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Bryant, C., Yuan, Z., Qorib, M. R., Cao, H., Ng, H. T., & Briscoe, T. (2023). Grammatical Error Correction: A Survey of the State of the Art. *Computational Linguistics*, *49*(3), 643–701. doi: 10.1162/coli_a_00478

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (str. 4171–4186). doi: 10.18653/v1/N19-1423

Dobrovoljc, K., Krek, S., & Erjavec, T. (2015). Leksikon besednih oblik Sloleks in smernice njegovega razvoja. V V. Gorjanc, P. Gantar, I. Kosem & S. Krek (ur.), *Slovar sodobne slovenščine: problemi in rešitve* (str. 80–105). Ljubljana: Znanstvena založba Filozofske fakultete Univerze v Ljubljani. Pridobljeno s https://ebooks.uni-lj.si/ZalozbaUL/catalog/view/15/47/489

Dobrovoljc, K., Krek, S., Holozan, P., Erjavec, T., Romih, M., Arhar Holdt, Š., Čibej, J., Krsnik, L., & Robnik-Šikonja, M. (2019). *Morphological lexicon Sloleks 2.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1230

Erjavec, T. (2017). MULTEXT-East. V *Handbook of Linguistic Annotation* (str. 441–462). Springer.

Erjavec, T., Fišer, D., Ljubešić, N., Ferme, M., Borovič, M., Boškovič, B., Ojsteršek, M., & Hrovat, G. (2019). *Corpus of Academic Slovene KAS 1.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1244

Fang, T., Yang, S., Lan, K., Wong, D. F., Hu, J., Chao, L. S., & Zhang, Y. (2023). Is ChatGPT a highly fluent grammatical error correction system? A comprehensive evaluation. *ArXiv*. doi: 10.48550/arXiv.2304.01746

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *International conference on machine learning*, 1050–1059.

Gantar, P., Bon, M., Gapsa, M., & Holdt, Š. A. (2023). Šolar-Eval: Evalvacijska množica za strojno popravljanje jezikovnih napak v slovenskih besedilih. *Jezik in slovstvo*, *68*(4), 89–108. doi: 10.4312/jis.68.4.89-108

Jurafsky, D., & Martin, J. H. (2024). *Speech and Language Processing (3rd ed. draft)*. Pridobljeno s https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Krek, S., Holdt, Š. A., Erjavec, T., Čibej, J., Repar, A., Gantar, P., Ljubešić, N., Kosem, I., & Dobrovoljc, K. (2020). Gigafida 2.0: The reference corpus of written standard Slovene. In N. Calzolari et al. (Eds.), *Proceedings of the Twelfth language resources and evaluation conference, LREC 2020, Marseille, France* (str. 3340–3345). The European Language Resources Association (ELRA).

Krek, S. (2023). Language Report Slovenian. In *European Language Equality: A Strategic Agenda for Digital Language Equality* (str. 211–214). Cham: Springer International Publishing.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*. doi: 10.48550/arXiv.1907.11692

Ljubešić, N., & Dobrovoljc, K. (2019). What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian. *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, 29–34. doi: 10.18653/v1/W19-3704

Miok, K., Škrlj, B., Zaharie, D., & Robnik-Šikonja, M. (2022). To BAN or not to BAN: Bayesian attention networks for reliable hate speech detection. *Cognitive Computation*, *14*(1), 353–371.

Mokotar, R. (2023). *Obvladovanje slovničnih napak v šolskih pisnih izdelkih z metodami za obdelavo naravnega jezika* (Diplomska naloga). Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Pridobljeno s https://repozitorij.uni-lj.si/IzpisGradiva.php?id=144932&lang=slv

Pančur, A., Erjavec, T., Ojsteršek, M., Šorn, M., & Blaj Hribar, N. (2020). *Slovenian parliamentary corpus (1990-2018) siParl 2.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1300

Popič, D. (2014). Revising translation revision in Slovenia. *New Horizons in Translation Research and Education 2,* 72–89. University of Eastern Finland Joensuu.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Rizvič, M. (2020). *Avtomatsko postavljanje ločil v surovem tekstu* (Magistrsko delo). Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Pridobljeno s https://repozitorij.uni-lj.si/IzpisGradiva.php?id=117687&lang=slv

Romih, M., & Holozan, P. (2002). Infrastruktura za razvoj jezikovnih tehnologij-korpus FIDA in sistem ASES. V T. Erjavec, J. Žganec Gros (ur.), *Jezikovne tehnologije, 14.–15. oktober, Ljubljana, Slovenija* (str. 166). Pridobljeno s http://nl.ijs.si/isjt02/zbornik/sdjt02-D02amebis.pdf

Rozovskaya, A., Roth, D., & Srikumar, V. (2014). Correcting grammatical verb errors. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (str. 358–367).

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., …, Azhar, F., idr. (2023). Llama: Open and efficient foundation language models. *ArXiv*. doi: 10.48550/arXiv.2302.13971

Ulčar, M., & Robnik-Šikonja, M. (2021a). SloBERTa: Slovene monolingual large pretrained masked language model. *Proceedings of Slovenian KDD Conference, SiKDD 2021, part of Information Society*.

Ulčar, M., & Robnik-Šikonja, M. (2021b). *Slovenian RoBERTa contextual embeddings model: SloBERTa 2.0*, Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1397

Ulčar, M., & Robnik-Šikonja, M. (2023). Sequence to sequence pretraining for a less-resourced Slovenian language. *Frontiers in Artificial Intelligence*, *6*. doi: 10.3389/frai.2023.932519

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Velikonja, N. (2021). *Segmentacija in postavljanje končnih ločil v slovenskih stavkih z modeli tipa BERT* (Diplomska naloga). Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Pridobljeno s https://repozitorij.uni-lj.si/IzpisGradiva.php?id=130323&lang=slv

Wu, H., Wang, W., Wan, Y., Jiao, W., & Lyu, M. (2023). ChatGPT or Grammarly? Evaluating ChatGPT on grammatical error correction benchmark. *ArXiv.* doi: 10.48550/arXiv.2303.13648

Zhang, S., Huang, H., Liu, J., & Li, H. (2020). Spelling Error Correction with Soft-Masked BERT. V D. Jurafsky, J. Chai, N. Schluter & J. Tetreault (ur.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, *July 2020* (str. 882–890). Association for Computational Linguistics. Pridobljeno s https://aclanthology.org/2020.acl-main.pdf

The importance of realistic evaluation: an example of correcting Slovene grammatical case and number with large language models

Frequent grammar errors in standard Slovene include using an incorrect grammatical conjugation or number. Using the large language model SloBERTa, we have developed a new methodology for the machine detection of such problems and tested it on incorrect use of the accusative instead of the genitive case and the plural instead of the dual. We applied standard natural language processing tools for Slovenian to evaluate and modify word forms in the input sentences, such as morphosyntactic tagger CLASSLA-Stanza and Slovenian word form lexicon Sloleks. The proposed corrections are based on word form statistics when using masked word prediction with a large language model. Due to the lack of sufficient training data, we trained the prediction models on synthetically generated errors. We first evaluated the performance of machine correction on synthetic data and the Lektor corpus, and later on a newly developed evaluation dataset Šolar-Eval. The evaluation on the first two datasets showed the excellent performance of the developed methodology (more than 90% of detected synthetically introduced errors), while with Šolar-Eval it had a far worse performance (only 29.5% of the problems with the genitive-accusative grammatical case were detected, and just 11.4% of those with the dual-plural grammatical number). Overall, the results show the danger of overfitting to datasets and the importance of evaluating on purposefully designed authentic datasets, which are still rare for Slovene.

Keywords: grammatical error correction, grammatical case, grammatical number, large language models, evaluation