SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 853

MODEL I SKUP PODATAKA ZA PREVOĐENJE ČAKAVSKOG NARJEČJA NA STANDARDNI HRVATSKI JEZIK

Florijan Sandalj

Zagreb, srpanj 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 853

MODEL I SKUP PODATAKA ZA PREVOĐENJE ČAKAVSKOG NARJEČJA NA STANDARDNI HRVATSKI JEZIK

Florijan Sandalj

Zagreb, srpanj 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 3. ožujka 2025.

DIPLOMSKI ZADATAK br. 853

Pristupnik:	**Florijan Sandalj (0036530775)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Model i skup podataka za prevođenje čakavskog narječja na standardni hrvatski jezik**

Opis zadatka:

Čakavsko narječje jedno je od tri narječja hrvatskog jezika i govori se u Istri, na hrvatskim otocima i uskom obalnom području Primorja i Dalmacije. Postoji tek nekolicina skupova podataka i radova u području obrade prirodnog jezika koji se bave čakavskim narječjem. Cilj ovog diplomskog rada je omogućiti automatskom prevođenje s čakavskog narječja na standardni hrvatski jezik. Pritom je potrebno izgraditi skup podataka koji sadrži rečenice na čakavskom narječju te njihov prijevod na standardni hrvatski jezik. Podatke bi označavali dobrovoljci, izvorni govornici narječja uz pomoć aplikacije ranije razvijene u okviru diplomskog projekta. Nakon što su podaci prikupljeni i provjereni, bit će korišteni za učenje i testiranje modela za zadatak prevođenja teksta s čakavskog u standardni hrvatski jezik. U radu je potrebno teoretski opisati suvremene pristupe temeljene na dubokom učenju za prevođenje dijalekata i rjeđih jezika na standardne jezike. U praktičnom dijelu, potrebno je fino podesiti za prevođenje s čakavskog narječja na standardni hrvatski jezik transformerski jezični model BERTić koji je specifično prednaučen na tekstovima balkanskih jezika, uključujući i hrvatski. Izgrađeni model potrebno je vrednovati na odvojenom dijelu izgrađenog skupa podataka te izvijestiti o rezultatima modela, uključujući kvalitativni dojam te uobičajene metrike korištene u sličnim zadacima prevođenja (npr. BLEU, METEOR).

Rok za predaju rada: 4. srpnja 2025.

Zahvaljujem svojoj obitelji i prijateljima na podršci i motivaciji tijekom studija. Također, zahvaljujem svom mentoru izv. prof. dr. sc. Alanu Joviću na mentorstvu tijekom studija te na korisnim savjetima i smjernicama tijekom izrade diplomskog rada.

Sadržaj

1. Uvod

Ovo poglavlje služi kao uvod u tematiku kojom se bavi ovaj diplomski rad. Ukratko će biti objašnjeno područje obrade prirodnog jezika, čime se bavi i zašto je važno te će biti uveden i problem standardizacije dijalekta. Nakon toga, slijedi nekoliko riječi o dijalektu koji se obrađuje u ovom radu, čakavskom narječju. Za kraj, tu je i pregled literature koja se bavi usko vezanom tematikom.

1.1. Obrada prirodnog jezika

Obrada prirodnog jezika (engl. *Natural Language Processing*, skraćeno NLP) je područje koje objedinjuje računalnu znanost, umjetnu inteligenciju i lingvistiku. Njena primarna zadaća je omogućiti računalima da razumiju i obrađuju informacije sadržane u ljudskom (prirodnom) jeziku te je, kao takva, usko povezana s područjima poput dohvaćanja informacija (engl. *Information Retrieval*), reprezentacije znanja (engl. *Knowledge Representation*) i računalne lingvistike (engl. *Computational Linguistics*). Izazovi u obradi prirodnog jezika proizlaze iz činjenice da je prirodni jezik izuzetno kompleksan, često dvosmislen i ovisan o kontekstu. Potpuno razumijevanje jedne rečenice nerijetko zahtijeva poznavanje širokog konteksta iz kojeg je rečenica nastala, te dodatno i posjedovanje nekog temeljnog općeg znanja. Zbog svega navedenog, potpuno razumijevanje prirodnog jezika smatra se UI-potpunim (engl. *AI-complete*) problemom, što znači da se ne može riješiti specifičnim algoritmom, već zahtijeva opću umjetnu inteligenciju. Područje obrade prirodnog jezika obuhvaća širok spektar zadataka koji se često isprepliću i nadopunjuju. Po vrsti problema koji se rješava, područje se može podijeliti na:

- **Sintaksu**—zadaci koji se fokusiraju na gramatiku, odnosno strukturu riječi i rečenica, pravila po kojima riječi nastaju, kako se mijenjaju i kombiniraju u rečenice. Uključuje zadatke poput korjenovanja, lematizacije, parsiranja i označavanja dije-

lova govora (engl. *part-of-speech tagging*).

- **Semantiku**—zadaci koji su fokusirani na značenje riječi u kontekstu u kojem se nalaze. Uključuje zadatke poput prepoznavanja imenovanih entiteta (engl. *named entity recognition*) i razlučivanja smisla riječi (engl. *word sense disambiguation*).

- **Pragmatički i zadaci više razine**—zadaci koji se fokusiraju na razumijevanje konteksta čitavih rečenica ili većih tekstualnih cjelina. To su zadaci poput analize sentimenta, sažimanja teksta i generiranja prirodnog jezika.

Osim po vrsti zadatka, područje analize prirodnog jezika može se podijeliti i po metodologiji kojom se pristupa rješavanju zadataka, i to na:

- **Pravila i simboličke metode**—pristup koji se oslanja na ručno definirana pravila, dominantan u ranim danima obrade prirodnog jezika.

- **Statističke metode**—pristup koji se temelji na probabilističkim modelima. Ovaj pristup je postao dominantan s pojavom velikih korpusa teksta i napretkom u statističkom učenju.

- **Duboko učenje**—pristup koji koristi duboke neuronske mreže za rješavanje zadataka obrade prirodnog jezika i trenutačno dominira područjem, posebno uz nedavni razvoj transformera i velikih jezičnih modela.

Zadatak koji se obrađuje u ovom radu je standardizacija dijalekta, što je veoma slično strojnom prevođenju, a pristup koji se koristi je duboko učenje.

1.2. Normalizacija teksta

Normalizacija teksta obuhvaća sve metode zamjene van-standardnih oblika riječi njihovim standardnim ekvivalentima. Ti postupci korisni su kao korak pripreme podataka kod mnogih zadataka obrade prirodnog jezika jer povećavaju homogenost podataka i smanjuju utjecaj šuma [1]. Primjerice, dva prominentna područja primjene normalizacije teksta su normalizacija povijesnih tekstova [2], koji dolaze iz različitih razdoblja i jezik kojim su pisani je u međuvremenu evoluirao, i normalizacija teksta koji generiraju korisnici (engl. *user-generated content*, UGC) [3], koji karakteriziraju skraćeni oblici ri-

ječi, uporaba žargona te pogreške u pisanju. Pored navedenih, normalizacija dijalekata ostaje relativno neistraženo područje.

1.3. Standardizacija dijalekata

Standardni jezik je oblik jezika koji je prošao značajnu normizaciju svoje gramatike, leksikona, sustava pisanja ili drugih obilježja zbog čega se među srodnim oblicima istog jezika ističe kao onaj s najvišim statusom. Najčešće je to službeni oblik jezika jedne države [4].

U slavenskoj dijalektologiji, pa samim time i u hrvatskoj, pojam dijalekt odnosi se na govor određenog područja ili skupine ljudi [5]. U stranoj literaturi, pojam dijalekt koristi se u širem smislu i obuhvaća više regionalnih oblika jezika. Pojam koji u slavenskom jezikoslovlju obuhvaća više dijalekata je narječje. Na primjer, u Hrvatskoj postoje tri narječja: štokavsko, kajkavsko i čakavsko. Svako od tih narječja obuhvaća brojne dijalekte koji su vezani uz neko geografsko područje ili skupinu ljudi. Sukladno tome, i budući da pojam *standardizacija dijalekta* potječe iz engleskog jezika, treba napomenuti da se pritom ne misli na strogi pojam dijalekta definiranog u hrvatskoj dijalektologiji, već je definicija nešto slobodnija i "dijalekt" može obuhvaćati i nekoliko "pravih" dijalekata, pa i čitavo narječje.

Kao što je već rečeno, standardizacija dijalekata relativno je slabo zastupljena u obradi prirodnog jezika, s tek nekolicinom kvalitetnih i temeljitih radova. Jedan od faktora je zasigurno nedostupnost dovoljne količine podataka, a moguće je i da se radi o pitanju isplativosti rješavanja tog problema i njegovoj primjeni u praksi/industriji.

1.4. Čakavsko narječje

Čakavsko narječje (čakavština, čakavica) jedno je od tri narječja hrvatskog jezika, uz štokavsko i kajkavsko. Naziv, kao i kod druga dva narječja, potječe od oblika upitne zamjenice *ča*. Čakavsko narječje u Hrvatskoj rasprostranjeno je u Istri, Kvarneru, otocima te uskom obalnom području Primorja i Dalmacije. Narječje se razvilo iz iste osnove kao i kajkavsko narječje, zapadnoštokavsko narječje i slovenski dijalekti te s njima dijeli mnoge srodnosti [6]. Također, ukupno se malo promijenilo od početnog oblika, pa se iz

tog razloga smatra arhaičnim. Neke od glavnih karakteristika koje obilježavaju čakavsko narječje su:

- upitne zamjenice *ča* i *zač*
- ikavski i ikavsko-ekavski govor
- kondicional *bin-biš-bimo, bimo*
- adrijatizmi, kao prijelaz *-m* u *-n* na kraju relacijskog morfema (moram—*moran*, čujem—*čujen*), i prijelaz *lj* u *j* (ljubav—*jubav*, polje—*poje*)
- arhaizmi (tipično na sjeveru) i romanizmi (svijećnjak—*kandelabar*)

Čakavsko je narječje sastavljeno od brojnih mjesnih govora od kojih svaki ima svoje posebnosti, a koji su grupirani u dijalekte. Postoji nekoliko podjela po više kriterija (akcentuacija, refleksi jata) u kojima se često pojavljuju sjeverno, srednje i južnočakavski, buzetski i lastovski.

1.5. Obrada prirodnog jezika i slavenska narječja

Kao što je već spomenuto, u polju standardizacije dijalekata postoji nekolicina radova uopće, a još manje onih koji se bave slavenskim jezicima. Međutim, jedan od opširnijih radova koji eksperimentira s različitim pristupima standardizaciji različitih dijalekata, uz finski, norveški i švicarski koriste i korpus slovenskog jezika. U radu [7] često korišteni skup podataka COPA preveden je uz pomoć izvornih govornika na tri dijalekta slovenskog, hrvatskog i srpskog jezika.

2. Prikupljanje podataka

Budući da se u ovom radu zadatku strojne standardizacije čakavskog narječja ne pristupa ručnim definiranjem pravila koja model treba slijediti (engl. *expert rules-based approach*), već se koriste duboki modeli, od interesa nisu gramatika i specifičnosti narječja, već je cilj prikupiti što više kvalitetnih podataka. Podaci koji se traže su riječi i rečenice na čakavskom narječju s odgovarajućim prijevodom na standardni hrvatski jezik. U ovom poglavlju bit će predstavljeni izvori podataka za ovaj diplomski rad.

2.1. Dobrovoljni anotatori

Jedna od prvih ideja kod izrade ovog rada bila je obratiti se institucijama koje se bave očuvanjem čakavskog narječja i pokušati okupiti dobrovoljce koji poznaju i govore jedan od dijalekata te bi bili voljni sudjelovati u prikupljanju podataka. U tu svrhu čak je izrađena i jednostavna web-stranica na kojoj bi dobrovoljci mogli prevoditi rečenice s čakavskog na standardni jezik i obratno. Nažalost, zbog manjka interesa ova se ideja nikad nije realizirala, tako da je bilo potrebno osloniti se na postojeće podatke.

2.2. Čakavski rječnik

Prvi sljedeći izvor podataka bio je Čakavski rječnik [8], javno dostupni korpus riječi i fraza s prijevodima i pojašnjenjima koji su pisali i dopunjavali autohtoni govornici, a pokriva dijalekt kvarnerskog kraja (Kastav, Grobnik, Kostrena, Krk itd.). Web stranica ne nudi opciju preuzimanja rječnika u tabličnom formatu, pa je tekst sa stranice trebalo predobraditi. Rječnik sadrži fond od 2054 riječi, ali neke imaju više standardnih i dijalektnih oblika, dok se na nekim mjestima uz prijevod pojavljuje objašnjenje koje je trebalo ukloniti. Nakon ručnog prepravljanja rječnik je sadržavao 2587 riječi i prijevoda.

2.3. DIALECT-COPA

Već spomenuti skup podataka DIALECT-COPA nastao je prevođenjem skupa podataka COPA (engl. *Choice of Plausible Alternatives*) [9] koji je namijenjen zadatku prepoznavanja uzročno posljedičnih veza u rečenicama. Skup podataka sadrži 1000 primjera, gdje se svaki primjer sastoji od tri rečenice: jedne premise i dva uzroka ili posljedice premise, gdje model treba prepoznati koja je od te dvije ispravna posljedica ili uzrok. Ispod se nalazi jedan primjer na hrvatskom i čakavskom dijalektu.

```
premisa: Dizao sam i spuštao prekidač za svjetlo.
  posljedica_1: Svjetlo se polako ugasilo.
  posljedica_2: Svjetlo je treperilo.
premisa: San diguva i kaleva šalter za svetlo.
  posljedica_1: Svetlo se je pomalon zagasilo.
  posljedica_2: Svetlo je treputalo.
```

Skup podataka COPA već je ranije preveden na hrvatski jezik [10], a u [7] je preveden na čakavski dijalekt istarskog grada Žminja. Podaci za učenje i validaciju (500 primjera) javno su dostupni, dok je testne podatke moguće dobiti na zahtjev kako bi se spriječilo njihova curenje u skupove za učenje velikih jezičnih modela, što bi učinilo skupove podataka COPA-HR i COPA-DIALECT neupotrebljivima za testiranje njihovih performansi. Čak i nakon dobivanja pristupa testnim podacima, njihova iskoristivost u ovom radu bila je ograničena, budući da su se testiranje i evaluacija modela odvijali na *online* platformama gdje je pametnije bilo ne učitati te podatke.

2.4. Mići princ

"Mići princ text and speech dataset" [11] skup je podataka koji je nastao prevođenjem poznate knjige *Mali princ* (fr. *Le Petit Prince*) [12] na čakavske dijalekte. U projektu je sudjelovalo više prevoditelja, pa skoro svaki lik u knjizi govori različitim mikro-dijalektom. Skup podataka objavljen je kao audio knjiga te sadrži i audio zapise cijelog teksta, ali to je nešto čime se ovaj rad ne bavi. Ukupno, skup podataka sadrži 11591 riječi. Skup podataka ne sadrži odgovarajuće rečenice na standardnom hrvatskom jeziku, pa su one prikupljene iz jednog od javno dostupnih izdanja knjige u PDF formatu [13]. Budući da

prijevod na čakavski nije rađen riječ-po-riječ, rečenicu-po-rečenicu, podatke je trebalo ručno pregledati te ukloniti dijelove koji nemaju pripadajući prijevod.

2.5. Zbirka pjesama *Ni pjesme za te*

U pokušaju pronalaska dobrovoljaca za prikupljanje podataka kontaktirana je glavna prevoditeljica na projektu DIALECT-COPA, dr. sc. Tea Perinčić, koja je ljubazno ponudila ustupiti svoju tada neobjavljenu zbirku pjesama pod nazivom *Ni pjesme za te*. Pjesme su napisane na čakavskom narječju i prevedene na standardni hrvatski. Zbirka sadrži 56 pjesama, 1108 stihova, 4427 riječi u čakavskoj verziji te 4369 riječi na standardnom hrvatskom jeziku. Budući da prijevod očigledno nije rađen riječ-po-riječ, ustanovljeno je da je najmanja cjelina koja može poslužiti za uparivanje čakavske verzije i standardnog prijevoda upravo stih.

3. Teorijska pozadina

U ovom poglavlju bit će objašnjena teorijska osnova korištenih metoda i modela.

3.1. Neuronska obrada prirodnog jezika

U neuronskoj obradi prirodnog jezika (engl. *Neural Natural Language Processing*) na neki se način iskorištava arhitektura neuronskih mreža i njihova sposobnost da efikasno nauče veliki broj parametara kroz algoritam propagacije pogreške unatrag. Statistička obrada prirodnog jezika razdoblje je koje je prethodilo neuronskoj, ali je imalo jedan veliki nedostatak, a to je da je zahtijevalo složen proces osmišljavanja i odabira značajki (engl. *feature engineering*). 2012. godine je u obradi prirodnog jezika uveden pristup dubokog učenja, nakon što su duboke neuronske mreže postigle značajan uspjeh u zadacima prepoznavanja objekata na slikama i prepoznavanja govora [14]. Ubrzo su metode dubokog učenja počele nadmašivati statističke metode i postavljati nove standarde u području.

Neki sustav obrade prirodnog jezika prima riječ ili niz riječi i vraća (jednu ili niz) oznaku klase ili novi niz riječi. Da bi primijenili metode dubokog učenja na ovakve zadatke, potrebno je riješiti dva ključna problema:

1. Kodiranje—pretvoriti rečenicu prirodnog jezika u oblik pogodan za duboki model
2. Generiranje—proizvesti izlaz koji se traži (oznaka klase ili rečenica)

Imajući na umu ova dva zadatka, bit će objašnjeno na koje se sve načine oni rješavaju, koje su prednosti i nedostatci tih pristupa i kako se oni odnose na problematiku ovog rada.

3.1.1. Kodiranje neovisno o kontekstu

Kodiranje služi za pretvaranje riječi ili rečenica u numerički oblik (vektor) koji u sebi ima ugrađene informacije koje nama nisu vidljive, ali ih duboki model može iskoristiti. Zato se kodirani oblik također naziva ugradnjom ili vektorskom reprezentacijom riječi (engl. embedding) te se govori o ugradnjama riječi ili rečenica (engl. word/sentence embedding).

Word2Vec

Najkorišteniji algoritam učenja ugradnji riječi neovisnih o kontekstu je Word2Vec [15], koji koristi dva pristupa učenju: CBOW (engl. *Continuous Bag of Words*) i Skip-Gram. Za oba pristupa potreban je veći korpus teksta, ali on ne treba biti označen—Word2Vec se oslanja na ideju da se riječ može prepoznati po okolini u kojoj se nalazi. Kod pristupa CBOW neuronska mreža na ulazu dobiva informacije o riječima koje se nalaze u okruženju ciljne riječi, a na izlazu treba predvidjeti ciljnu riječ. Skip-Gram pak prima ciljnu riječ na ulazu, a na izlazu predviđa nalazi li se neka riječ u okolini ulazne riječi ili ne. U oba pristupa će, uz dovoljno podataka i kroz dovoljno iteracija, mreža za svaku riječ u korpusu naučiti upotrebljivu vektorsku reprezentaciju. Rezultat učenja je dakle rječnik koji svakoj riječi iz korpusa pridružuje ugradnju. Upravo se zato kaže da je takva ugradnja neovisna o kontekstu—stvar je u tome da se kontekst (okolina riječi) koristi samo pri učenju ugradnje. U primjeni, nakon završetka učenja, vektorska reprezentacija riječi uvijek će biti ista bez obzira na kontekst u kojem se riječ nalazi.

To je i glavni nedostatak ovog pristupa. Jezik je kompleksan i dvosmislen, a značenje riječi često se mijenja ovisno o njenoj okolini. Čim se radi s rečenicama, poželjno je da model dobije informaciju o kontekstu u kojem se riječ nalazi.

3.1.2. Reprezentacije obogaćene kontekstom

Tipični ulaz nekog modela obrade prirodnog jezika je niz riječi (rečenica) duljine T. Riječi se kodiraju u ugradnje fiksne duljine D (npr. 300), a više se rečenica obrađuje istovremeno u skupini (engl. *batch*) veličine B. Ovisno o zadatku, želimo:

- Klasifikacija—cilj je rečenicu predstaviti ugradnjom fiksne duljine koja se prosljeđuje klasifikatoru

- Označavanje sekvenci—cilj je svaku riječ predstaviti ugradnjom fiksne duljine koja je sadrži informacije o kontekstu unutar rečenice
- Generiranje teksta—cilj je rečenicu predstaviti ugradnjom fiksne duljine iz koje će dekoder proizvesti novu rečenicu

Dakle, u nekom je trenutku rečenicu varijabilne duljine potrebno svesti na fiksnu reprezentaciju. Možemo koristiti jednostavne funkcije poput prosjeka, zbroja, maksimuma ili težinskog prosjeka. Međutim, postoje dva problema s takvim funkcijama: (1) invarijantne su na redoslijed riječi u rečenici, i (2) kodiraju samo cijelu rečenicu, a ne i pojedine riječi.

Povratni modeli

Ako je s fiksna reprezentacija rečenice, a x_i su kontekstno neovisne ugradnje pojedinih riječi, tada se traži funkcija f koja će iz niza ugradnji vratiti reprezentaciju s:

$$s = f(x_0, x_1, \dots, x_T) \tag{3.1}$$

koja također vraća i kontekstno obogaćene ugradnje pojedinih riječi:

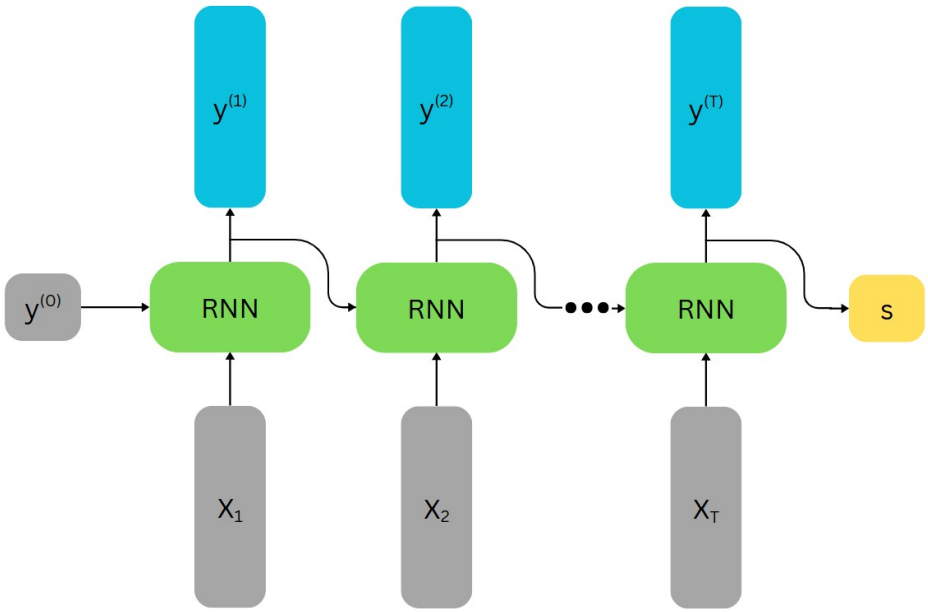
$$y_i = f(x_0, x_1, \dots, x_T) \quad i = 0, 1, \dots, T \tag{3.2}$$

Upravo taj zadatak rješavaju povratni modeli (engl. *recurrent neural networks*, RNN). Povratnom modelu se na ulaz postepeno dostavljaju ugradnje riječi, a izlaz nakon svakog koraka je kontekstno obogaćena reprezentacija te riječi. Na ulazu se uz ugradnju riječi dostavlja i izlaz povratnog modela iz prethodnog koraka (od tuda i naziv povratni modeli), što omogućuje modelu da znanja o prethodnim riječima prenese u reprezentaciju svake sljedeće riječi. Rad povratnog modela prikazan je na slici 3.1.

Iako je ovo veliki napredak u odnosu na jednostavne funkcije agregacije, i povratni modeli imaju svoje nedostatke.

- Potreba za dugoročnim pamćenjem—što su dvije riječi udaljenije u ulaznom tekstu, to je veća vjerojatnost da će model "zaboraviti" informaciju koju nosi prva riječ kada dođe do druge.

- Eksplodirajući/iščezavajući gradijent—tijekom učenja algoritmom propagacije pogreške unatrag, gradijent se višestruko množi s težinama dubokog modela. Ovisno o svojstvenim vrijednostima matrice težina, gradijent može težiti u beskonačnost ili nulu.



Slika 3.1. Jednostavan povratni model

Različite arhitekture predložene su kako bi se doskočilo navedenim problemima, a dvije najpoznatije su LSTM (engl. *Long Short-Term Memory*) i GRU (engl. *Gated Recurrent Unit*).

LSTM

Arhitektura LSTM, prvi put u cijelosti opisana u [16], koristi sljedeće principe kako bi unaprijedila performance jednostavnih povratnih modela:

- Dvojno stanje ćelije—podjela odgovornosti na izlaznu komponentu koja se brine o izlazu i komponentu stanja koja je zadužena za memoriju.
- Mehanizam vrata (engl. *gates*) kontrolira protok informacija kroz ćeliju
- Stabilnost algoritma unatražnog prijenosa zbog aditivnog ažuriranja stanja

U svakom koraku ćelije LSTM-a na ulazu prima ugradnju riječi $x^{(t)}$, izlaz iz prethodnog koraka $h^{(t-1)}$ i stanje ćelije (memoriju) iz prethodnog koraka $c^{(t-1)}$, a na izlazu vraća novo

stanje ćelije $c^{(t)}$ i izlaz $h^{(t)}$. ćelija LSTM-a koristi tri vrata za kontrolu protoka informacija, i to vrata novog ulaza 3.3 (engl. *input gate*), vrata zaboravljanja 3.4 (engl. *forget gate*) i izlazna vrata 3.5 (engl. *output gate*). Vrijednosti pojedinih vrata računaju se po sljedećim formulama:

$$g^{(t)} = \sigma(W_{gx}x^{(t)} + U_{gh}h^{(t-1)} + b_g) \tag{3.3}$$

$$f^{(t)} = \sigma(W_{fx}x^{(t)} + U_{fh}h^{(t-1)} + b_f) \tag{3.4}$$

$$o^{(t)} = \sigma(W_{ox}x^{(t)} + U_{oh}h^{(t-1)} + b_o) \tag{3.5}$$

Također, transformacija ulaza $\hat{c}^{(t)}$ računa se po formuli:

$$\hat{c}^{(t)} = \tanh(W_{cx}x^{(t)} + U_{ch}h^{(t-1)} + b_c) \tag{3.6}$$

Vrata LSTM-a rade kao propusnice. Ona množenjem bit-po-bit (*Hadamardov produkt*) propuštaju ili blokiraju prijenos određenog dijela informacije. Tako se novo stanje ćelije LSTM-a računa po formuli:

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + g^{(t)} \odot \hat{c}^{(t)} \tag{3.7}$$

gdje je $\odot$ Hadamardov produkt. Izlaz iz ćelije LSTM-a računa se kao:

$$h^{(t)} = o^{(t)} \odot \tanh(c^{(t)}) \tag{3.8}$$

Dakle, u svakom koraku ćelija LSTM-a na osnovu riječi koja se obrađuje i prethodnog izlaza odlučuje koje će informacije iz memorije zadržati, a koje odbaciti. Na sličan način odlučuje koje će ulazne informacije propustiti u memoriju. Konačno, odlučuje i koje će informacije iz memorije poslati na izlaz. Arhitektura LSTM može se proširiti u dvosmjerni Bi-LSTM (engl. *bidirectional LSTM*) model koji se zapravo sastoji od dva LSTM-a, jedan koji obrađuje riječi u rečenici s lijeva na desno, a drugi s desna na lijevo. Tako se dobiva dvostruka količina informacije (dva vektora), pa se oni spajaju na određen način, najčešće zbrajanjem ili konkatenacijom.

Ćelija GRU neće biti objašnjena u previše detalja, dovoljno je reći da radi na istom principu kao i LSTM, ali ne sadrži dva stanja ćelije, pa je arhitektura nešto jednostavnije s manje parametara. Iako su se arhitekture zasnovane na ćelijama LSTM i GRU pokazale kao unaprjeđenje u odnosu na jednostavne povratne modele, one i dalje imaju ograničene mogućnosti pamćenja dugoročnih ovisnosti. Jedan od razloga tomu je ograničena količina informacije koja stane vektor stanja ćelije, a drugi je taj što u trenutku procesiranja neke riječi model nema uvid "budućnost". Postoji doza neizvjesnosti u tome kako će se rečenica nastaviti, a model mora na neki način sve to uzeti u obzir prilikom enkodiranja, što dodatno opterećuje memoriju. Tu na red dolazi sljedeći važan koncept, a to je pažnja.

Mehanizam pažnje

Mehanizam pažnje (engl. *attention*) prvi je put predstavljen u [17] i pokazao se ključnim za daljnji razvoj u području. Zasniva se na ideja da se modelu omogući da "preleti" preko prethodnih riječi u rečenici i stavi fokus na one koje su relevantne za riječ koja se trenutačno obrađuje. Na taj se način rasterećuje memorija i efikasnije dodaje relevantna informacija izlazu.

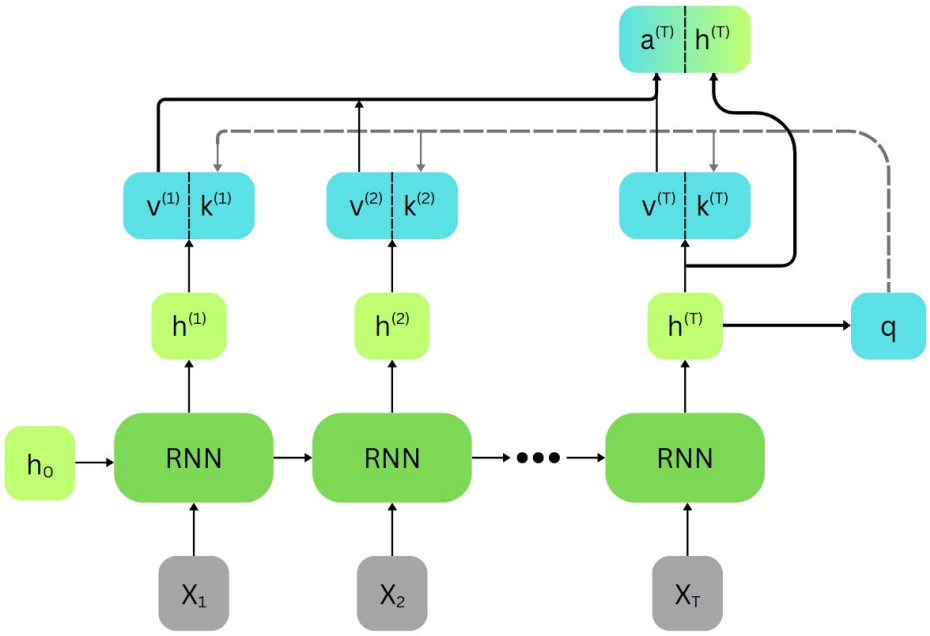
Neka je $q \in \mathbb{R}^q$ vektor upita, $K \in \mathbb{R}^{T \times k}$ matrica ključeva i $V \in \mathbb{R}^{T \times v}$ matrica vrijednosti. Ideja je podjela odgovornosti: q i K sadrže informacije potrebne za pretraživanje, odnosno pronalaženje relevantnih riječi. Da bi to bilo moguće, treba nam i funkcija energije (engl. *energy function*) $f_e : \mathbb{R}^q \times \mathbb{R}^k \to \mathbb{R}$ koja će za par (upit, ključ) vratiti skalarnu vrijednost koja predstavlja relevantnost. Kada se izračuna energija kroz sve korake (dimenzija T), dobiva se vektor $e \in \mathbb{R}^T$ koji se provlači kroz softmax: $\alpha = \text{softmax}(e)$. Vektor α sadrži težine kojima se množe vrijednosti iz V i tako konačno dobiva vektor pažnje:

$$a = \sum_{i=0}^{T} \alpha_i v_i \tag{3.9}$$

Skica mehanizma pažnje preko jednoslojnog povratnog modela prikazana je na slici 3.2. Prvi prijedlog mehanizma pažnje za funkciju energije koristio je skalarni umnožak vektora upita i vektora ključa podijeljen s korijenom dimenzije tih vektora, što je ostalo najkorištenija varijanta te se naziva Bahdanauova pažnja [17].

Implementacija pažnje u LSTM-ovima prilično je pojednostavljena. Za vektor upita koristi se stanje ćelije $q = c^{(T)}$, a ključevi i vrijednosti uzimaju se iz izlaza LSTM-a $k^{(t)} = v^{(t)} = h^{(t)}$. Tako se računanje energije svodi na:

$$e_t = \frac{c^{(T)} \cdot h^{(t)}}{\sqrt{dim_h}} \tag{3.10}$$



Slika 3.2. Mehanizam pažnje preko jednostavnog povratnog modela

3.1.3. Veliki jezični modeli

Veliki jezični modeli (engl. *large language models*, LLM) su duboke neuronske mreže koje se najčešće temelje na arhitekturi transformera. Učeni su na ogromnim skupovima (često neoznačenih) podataka kako bi skupili sveobuhvatno razumijevanje prirodnog jezika. Imaju jako puno parametara, ogromne kapacitete i pokazuju izvanredne performanse na zadacima obrade prirodnog jezika, od klasifikacije i označavanja pa do prevođenja, sažimanja i odgovaranja na pitanja.

Transformeri

Mehanizam pažnje unaprijedio je povratne modele pomogavši s dugotrajnim pamćenjem te lakšom propagacijom gradijenta. Prirodno se postavlja pitanje: zašto ne izgraditi model koji se u potpunosti oslanja na pažnju (engl. *fully attentional network*)? Jedino što

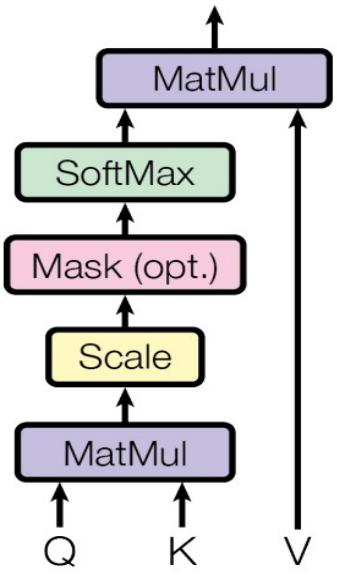
pažnji nedostaje je informacija o relativnoj poziciji riječi unutar rečenice. Ta se informacija može dodati na dva načina:

- Fiksnom ugradnjom—za ovaj pristup koriste se sinusne i kosinusne funkcije
- Učenje ugradnji—ugradnje se nasumično inicijaliziraju i uče s ostalim parametrima mreže

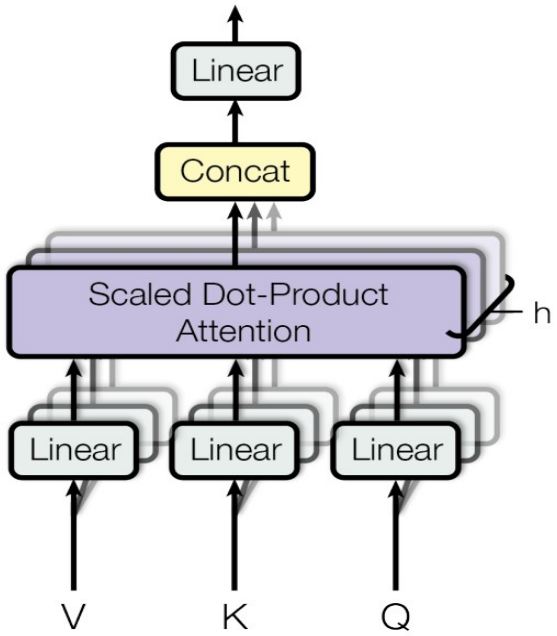
I to je sve što je potrebno za izgradnju modela koji se u potpunosti oslanja na pažnju. Najpoznatija arhitektura takvog modela naziva se Transformer i predstavljena je u [18], radu koji se smatra prekretnicom u obradi prirodnog jezika i jednim od pokretača nedavne "eksplozije" velikih jezičnih modela.

Pažnja u transformerima

Mehanizam pažnje kod transformera razlikuje se od onog kod LSTM-ova po tome što se pažnja računa za svaku riječ u rečenici, i to preko svih drugih riječi, za razliku od LSTM-a, gdje se pažnja računala samo preko prethodnih riječi. To se može jednostavno zapisati matričnim umnoškom:



Slika 3.3. Koder transformera (preuzeto iz [18])



Slika 3.4. Dekoder transformera (preuzeto iz [18])

$$\text{Pažnja}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.11}$$

Gdje su Q, K i V matrice upita, ključeva i vrijednosti, a $d_k = d_q$ dimenzije ključeva i vrijednosti. Transformeri koriste pažnju s više "glava" (engl. *multi-head attention*) koja se implementira linearnom projekcijom ključeva, upita i vrijednosti u različite prostore (8 "glava" u izvornom radu), te se za svaki prostor pažnja računa zasebno. Rezultati svake "glave" se konkateniraju i ponovno projicira u konačni rezultat. Također, postoji iznimka u kojoj se pažnja ne smije računati preko riječi koje dolaze poslije određene riječi u rečenici. U tom slučaju, koristi se "maska" kojom se prekrivaju te riječi. Sve je to vidljivo na skicama slika 3.3. i 3.4.

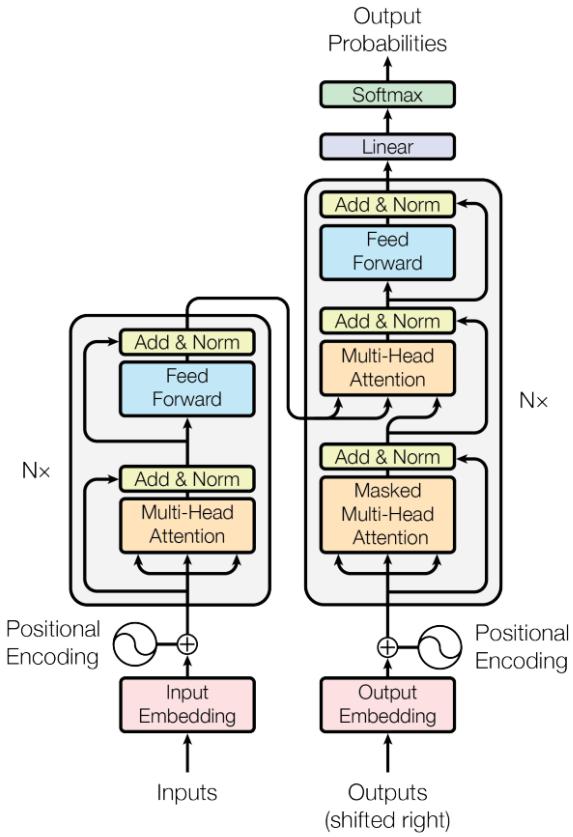


Figure 1: The Transformer - model architecture.

Slika 3.5. Arhitektura transformera (preuzeto iz [18])

Arhitekturu transformera čine dvije komponente: koder (engl. encoder) i dekoder (engl. decoder). Obje komponente započinju pretvaranjem ulaznih riječi u ugradnje i dodavanjem informacije o poziciji. Nakon toga, svaka se komponenta sastoji od više slo-

jeva (6 u izvornom radu) koji su arhitekturno jednaki, ali s različitim parametrima. Kod kodera, svaki se sloj sastoji od dva podsloja: mehanizma pažnje s više glava i potpuno povezanog unaprijednog sloja. Ulazi u mehanizam pažnje kodera su ugradnje riječi iz ulaza. Kod dekodera, svaki se sloj sastoji od tri podsloja: dva mehanizma pažnje s više glava i jednog potpuno povezanog unaprijednog sloja. Ulazi u prvi mehanizam pažnje dekodera su ugradnje riječi koje je dekoder do tada generirao (ovdje se koristi maska za prekrivanje tokena koje određena riječ nije vidjela kada je bila generirana). Kod drugog mehanizma pažnje, ključevi i vrijednosti dobivaju se od izlaza iz dekodera, dok su upiti dobivaju od izlaza iz prvog mehanizma pažnje dekodera. Izlazi iz dekodera prolaze kroz linearni sloj i aktivaciju *softmax* nakon čega se dobiva distribucija vjerojatnosti po riječima iz vokabulara. Skica cijele arhitekture prikazana je na slici 3.5.

BERT

Jedna od poznatijih primjena transformerske arhitekture je BERT (engl. *Bidirectional Encoder Representations from Transformers*) [19]. BERT koristi samo koder transformera i uči ga na dva zadatka:

- Predviđanje skrivene riječi—(engl. *masked language modelling*)—modelu se daje rečenica u kojoj je udio riječi zamijenjen tokenom [MASK], model uči predviđati skrivene riječi
- Predviđanje sljedeće rečenice—model predviđa nalazi li se neka rečenica neposredno iza zadane ili ne

Rezultat je koder koji generira ugradnje obogaćene kontekstom koje se mogu koristiti za razne zadatke klasifikacije, označavanja, pa čak i generiranja teksta. U izvornom radu BERT je učen na engleskom korpusu od 3.3 milijarde riječi. Kako se pokazao izrazito uspješnim, počele su se pojavljivati inačice BERT-a učene na drugim jezicima. Inačica učena na 8 milijardi tokena bosanskog, hrvatskog, makedonskog i srpskog jezika nosi ime BERTić i predstavljena je u [10].

(m)BART

BART (engl. *Bidirectional and Auto-Regressive Transformers*) [20] je koder-dekoder model temeljen na arhitekturi transformera koji je učen na zadatku rekonstrukcije teksta.

Tijekom učenja u ulazne rečenice različitim tehnikama unesena je određena količina šuma, a model je učio ukloniti šum i rekonstruirati izvornu rečenicu. BART je izvorno učen na engleskom jeziku, da bi kasnije nastao i mBART (engl. *Multilingual BART*) [21] koji je učen na 25 jezika, te proširenje na dodatnih 25 jezika, uključujući i hrvatski, mBART-50 [22].

(m)T5

Model T5 (engl. *Text-to-Text Transfer Transformer*) [23] koristi arhitekturu transformera i uči ju na zadatku rekonstrukcije teksta (slično kao i BERT i BART), ali s bitnom razlikom: kod modela T5, umjesto da se maskiraju pojedini tokeni (riječi) u ulaznom tekstu, maskiraju se čitavi rasponi riječi. Cilj ovakvog pristupa je da se model bolje nauči generirati tekst. Ovaj model sve zadatke obrade prirodnog jezika (prevođenje, klasifikacija, generiranje teksta) svodi na "text-to-text" format, gdje su i ulazi i izlazi u tekstualnom obliku. Model mT5 [24] proširuje T5 učeći ga na mC4 korpusu [25] koji sadrži preko 100 jezika, uključujući i hrvatski.

GPT

GPT (engl. *Generative Pre-trained Transformer*) [26] je model temeljen na transformeru koji je osmišljen s ciljem boljeg razumijevanja i generiranja teksta. Arhitektura se sastoji od višeslojnog dekodera transformera (bez kodera) koji je učen na zadatku predviđanja sljedeće riječi u sekvenci. Model je prvi put predložen 2018., a do sada je razvijeno nekoliko inačica. Najnovija, pod nazivom GPT-4 [27], ima procijenjenih 1.7 bilijuna parametara [28].

3.1.4. Augmentacija podataka

Augmentacija podataka (engl. *data augmentation*) tehnika je koja se često koristi u području strojnog učenja, a služi za povećanje količine podataka dostupnih za učenje modela. Postoji više razloga zbog kojih bi se mogli odlučiti na augmentaciju podataka. Za početak, ovisno o zadataku, čak i prikupljanje neoznačenih podataka može predstavljati izazov. Ako su pak neoznačeni podaci dostupni i treba ih označiti, to gotovo uvijek predstavlja izazov. Ručno označavanje podataka često zahtijeva stručnost i koordinaciju anotatora, skupo je i dugotrajno. Augmentacija podataka omogućava da na se razne na-

čine stvore novi, sintetički primjeri koji možda neće imati jednaku vrijednost kao stvarni podaci, ali i dalje mogu poboljšati performanse i dati uvid u to kako bi model mogao raditi sa stvarnim podacima.

Augmentacija podataka izvorno je ideja korištena u računalnom vidu [29] koja se proširila i na ostale domene, pa tako i obradu prirodnog jezika. Jednostavne metode augmentacije uključuju zamjenu riječi sinonimima, permutaciju ili dodavanje riječi u rečenicu [30]. Naprednije metode koriste modele dubokog učenja, pa tako i velike jezične modele (LLM) za označavanje postojećih podataka ili generiranje novih. U [31] pokazalo se da, iako stvarni podaci ili oni koje su označavali ljudski anotatori rezultiraju značajno boljim performansama, augmentacija korištenjem velikih jezičnih modela i dalje može biti korisna kada postoji nedostatak stvarnih podataka.

3.1.5. Metrike evaluacije

U ovom poglavlju predstavljene su metrike koje se u radu koriste za evaluaciju modela. Iako se u obradi prirodnog jezika često koriste metrike BLEU (engl. *Bilingual Evaluation Understudy*) [32] i ROUGE (engl. *Recall-Oriented Understudy for Gisting Evaluation*) [33], one nisu prikladne za ovaj zadatak: metrika ROUGE razvijena je za evaluaciju sažimanja teksta i fokusira se na odziv, dok je metrika BLEU po nekim ocjenama prestroga, pogotovo kada se radi o morfološko bogatim jezicima [34]. Metrike koje se koriste u ovom radu su chrF i CER.

chrF

chrF je metrika koja mjeri sličnost između dvije sekvence oslanjajući se na F-mjeru na razini nizova znakova. Za početak se iz ciljne sekvence i sekvence koja se evaluira uzimaju svi podnizovi znakova duljine n (uz preklapanje). U izvornom radu koristi se $n = 6$. Zatim se računaju preciznost i odziv:

- **chrFP**—preciznost, odnosno postotak podnizova iz evaluirane sekvence koji imaju svoj par u ciljnoj sekvenci
- **chrFR**—odziv, odnosno postotak podnizova iz ciljne sekvence koji imaju svoj par u evaluiranoj sekvenci

nakon toga, mjera chrF računa se kao:

$$\text{chrF} = (1 + \beta^2)\frac{\text{chrFP} \cdot \text{chrFR}}{\beta^2\text{chrFP} + \text{chrFR}} \tag{3.12}$$

gdje parametar β određuje koliko je puta odziv važniji od preciznosti. Najčešće se koristi $\beta = 1$.

CER

CER (engl. *Character Error Rate*) je mjera koja se koristi za evaluaciju sličnosti između dvije sekvence. Temelji se na Levenshteinovoj udaljenosti. Levenshteinova udaljenost jedna je od najstarijih mjera sličnosti između dvije sekvence. Mjeri koliko je minimalno operacija nad jednim znakom potrebno da bi se jedna sekvenca izjednačila s drugom. Operacijama nad jednim znakom smatraju se brisanje, umetanje ili zamjena znaka. Nakon toga, CER se računa dijeljenjem Levenshteinove udaljenosti s duljinom ciljne sekvence.

4. Eksperimenti i rezultati

U ovom poglavlju bit će predstavljene postavke provedenih eksperimenata te njihovi rezultati. Svaka radnja koja je zahtijevala značajnije računalne resurse izvodila se preko platforme Kaggle [35] na dvije GPU jedinice NVIDIA Tesla T4 [36].

4.1. Podaci

Nakon ručne provjere podataka i dotjerivanja onih koji su sadržavali nepravilnosti ili redundancije, konačne brojke izgledale su ovako:

- COPA: 1500 primjera
- Mići Princ: 1530 primjera
- Pjesme: 1107 primjera
- Rječnik: 2323 primjera

Obzirom da je cilj bio fokusirati se na prevođenje rečenica, odlučeno je da se podaci iz čakavskog rječnika ne koriste u testiranju i validaciji, već samo u učenju. Ostala tri skupa podijeljena su u omjeru 81:9:10 na skupove za učenje, validaciju i testiranje. Ovako visoki postotak podataka u skupu za učenje posljedica je iznimno malog korpusa. Kada su se podaci spojili, dobili smo:

- Skup za učenje: 5673 primjera
- Skup za validaciju: 373 primjera
- Skup za testiranje: 414 primjera

Skup za validaciju korišten je za odabir hiperparametara i zadovoljavajućeg broja epoha

učenja. Kod konačnog testiranja, model je učen na uniji skupova za učenje i validaciju. Također, budući da je skup za testiranje sastavljen od tri veoma različita skupa podataka, u tablicama rezultata dodatno će biti prikazan i rezultat koji je model ostvario na podacima iz svakog pojedinačnog skupa (COPA, Mići Princ i Pjesme).

4.2. Augmentacija podataka

Kako bi se pokušalo kompenzirati za manjak podataka, isprobana je augmentacija podataka korištenjem velikih jezičnih modela. Za augmentaciju je odabran GPT-4.1 model. Korištena je tehnika "few-shot" upita, gdje se modelu uz opis zadatka daje i nekoliko primjera. U ovom slučaju korišten je 3-shot pristup, a primjeri su birani nasumično bez ponavljanja iz skupa za učenje. Jedan primjer upita izgledao je ovako:

```
Generiraj tri rečenice na čakavskom narječju te njihove prijevode
na standardni hrvatski jezik:
fuminanti -> šibice
plavuće -> plutajuće
Kamo biš tev zet moju ovcu -> Kamo želiš ponijeti moju ovcu
```

Korištene su zadane postavke modela (`temperature`=1.0, `top_p`=1.0, `max_tokens`=2048). Kvaliteta samih generiranih podataka bit će diskutirana kasnije, za sada treba samo napomenuti da su generirani odgovori imali prilično nekonzistentnu strukturu, te je bio blagi izazov pravilno izdvojiti generirane rečenice. Tu je možda postojala prilika za unaprjeđenje upita, odnosno davanja strogih uputa oko strukture odgovora, ali zbog ograničenih resursa nije bilo moguće provesti naknadne eksperimente. Unatoč tome, model je generirao po tri nova para rečenica za svaki upit, što je rezultiralo udvostručavanjem skupa za učenje (ukupno 11346 primjera).

4.3. Osnovni model i naivna arhitektura

Za trivijalnu osnovu koja služi kao minimum koji bi model trebao ostvariti koristi se nepromijenjena rečenica u dijalektu. Razlog zašto se ovo smije napraviti je upravo jer se ne radi o klasičnom zadatku prevođenja, već o standardizaciji dijalekta. Mnoge riječi i dijelovi riječi ne razlikuju se u dijalektu i standardnom jeziku, što je također nešto što

model treba naučiti. Rezultati osnovnog modela su prikazani u tablici 4.1. Zbog slabih performanci, ova se trivijalna arhitektura neće uspoređivati s ostalim modelima.

Skup	chrF	CER
Test	38.03	0.4297
COPA	31.97	0.5174
Mići Princ	38.69	0.4327
Pjesme	49.66	0.3009

Tablica 4.1. Rezultati osnovnog modela

Može se primijetiti da je najveća sličnost između dijalekta i standardnog jezika prisutna u skupu Pjesme, nešto manja u prijevodu Malog Princa te najmanja u skupu podataka COPA .

Također, ideja je bila konstruirati i naučiti jednostavan model koji bi poslužio kao još jedan osnovni model za usporedbu. Model koristi prethodno naučeni koder BERTić za ekstrakciju ugradnji riječi, te sloj LSTM-a s pažnjom kao dekoder. BERTić je u ovom slučaju zamrznut, odnosno njegovi parametri i posljedično ugradnje se ne mijenjaju. Učenje se provodi nad parametrima sloja LSTM i pažnjom. Model je učen na skupu za učenje i na proširenom skupu za učenje s augmentiranim podacima te je postigao izuzetno slabe rezultate, koji su prikazani u tablicama. Rezultati su odraz kompleksnosti zadataka obrade prirodnog jezika i pokazuju koliko se malo može napraviti "iz nule" i s malo podataka bez korištenja unaprijed naučenih modela za generiranje teksta.

Skup	chrF	CER
Test	20.58	1.2257
COPA	21.34	1.0999
Mići Princ	19.93	1.0711
Pjesme	20.79	1.6088

Tablica 4.2. Rezultati BERTić + LSTM + Attn

Skup	chrF	CER
Test	20.81	0.7092
COPA	22.26	0.6592
Mići Princ	19.89	0.6952
Pjesme	20.32	0.7963

Tablica 4.3. Rezultati BERTić + LSTM + Attn s augmentiranim podacima

4.4. mT5

Za testiranje mT5 modela korištena je njegova najmanja inačica, mT5-small [37], koja raspolaže s 300 milijuna parametara [24]. Razlog tomu je ograničena memorija dostupna na platformi Kaggle. Korišten je i pripadajući prethodno naučeni *MT5Tokenizer*. Stopa učenja postavljena je na $3e^{-4}$, a veličina grupe (engl. *batch size*) na 4 (ponovno zbog memorijskih ograničenja). učenjem i validacijom modela ustanovljeno je da se performanse modela poboljšavaju do čak 16. epohe, nakon čega kreću opadati. Nakon učenja na čitavom skupu za učenje dobiveni rezultati prikazani su u tablici 4.4.

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	43.73	0.3936	38.03	0.4297
COPA	41.22	0.4130	31.97	0.5174
Mići Princ	42.44	0.4247	38.69	0.4327
Pjesme	53.13	0.3243	49.66	0.3009

Tablica 4.4. Rezultati mT5-small

Ovi rezultati nisu poželjni, ali ne iznenađuju s obzirom da se radi o najmanjoj inačici modela i manjku podataka. Model je i dalje ostvario bolje rezultate od osnovnog pristupa na svim skupovima i metrikama osim CER na skupu Pjesme. Model je učeni na skupu podataka proširenog s augmentiranim primjerima. Rezultati su prikazani u tablici 4.5.

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	48.02	0.3623	38.03	0.4297
COPA	45.46	0.3865	31.97	0.5174
Mići Princ	46.52	0.3882	38.69	0.4327
Pjesme	58.17	0.2938	49.66	0.3009

Tablica 4.5. Rezultati mT5-small s augmentiranim podacima

Model i dalje ne daje sjajne rezultate, ali je vidljiv napredak s povećanjem količine podataka. Model sada nadmašuje osnovni model na svim skupovima i metrikama.

4.5. mBART

Koristi se najopćenitiju inačicu mBART-50 modela, mBART-50-large, koja nije fino podešena ni na jednom specifičnom zadatku, već je sam model prethodno naučen s namjenom prevođenja [22]. Korišten je pripadajući *mBART50Tokenizer* u kojemu je definirano `src_lang = HR_hr` i `tgt_lang = HR_hr`. Stopa učenja postavljena je na $3e^{-5}$, a veličina

grupe na 2 (memorijska ograničenja, a ovdje je korišten još veći model sa 610 milijuna parametara). Učenjem i validacijom ustanovljeno je da performanse modela rastu do 3. epohe, nakon čega stagniraju te opadaju. Dobiveni rezultati nakon učenja na čitavom skupu za učenje prikazani su u tablici

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	55.38	0.3112	38.03	0.4297
COPA	52.90	0.3337	31.97	0.5174
Mići Princ	54.38	0.3371	38.69	0.4327
Pjesme	63.90	0.2450	49.66	0.3009

Tablica 4.6. Rezultati mBART-50-large

Ovaj model daje bolje rezultate, što je i očekivano s obzirom na veličinu i namjenu. Model je također učen na proširenom skupu za učenje s augmentiranim primjerima, a rezultati su prikazani u tablici 4.7. Augmentirani podaci ne rade drastičnu razliku, ali ponovno poboljšavaju performanse modela po svim kriterijima.

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	58.66	0.2857	38.03	0.4297
COPA	55.85	0.3113	31.97	0.5174
Mići Princ	57.87	0.3095	38.69	0.4327
Pjesme	67.20	0.2183	49.66	0.3009

Tablica 4.7. Rezultati mBART-50-large s augmentiranim podacima

4.6. GPT-4

Konačno, na zadatku je testirana i GPT arhitektura, i to nova GPT-4.1 inačica. Iako je ovaj model nekoliko redova veličine veći od ostalih korištenih modela, GPT-4.1 se ne preuzima i pokreće lokalno, već se koristi putem OpenAI API-ja [38]. Također, posljedica ovakvog pristupa je nemogućnost učenja, odnosno u ovom slučaju finog podešavanja (engl. *fine-tuning*) modela. Umjesto toga, koristi se učenje kroz upite. To znači da se, ako je mišljenje da će to poboljšati performance modela, uz upit mogu priložiti i neki primjeri. Ti primjeri neće utjecati na model nakon što se upit izvrši, ali možda će trenutačno usmjeriti model prema željenom cilju. Takav se pristup naziva "few-shot" upit, za razliku od "zero-shot" upita, gdje modelu nisu dostavljeni primjeri, već samo upit. Na GPT-4.1 arhitekturi isproban je "zero-shot" i "3-shot" pristup. Primjer upita u "3-shot" pristupu izgleda ovako:

```
Prevedi sljedeće rečenice s čakavskog narječja na standardni hrvatski
po uzoru na sljedeće primjere:
Komet je pasa zuz mesec. -> Komet je prošao uz mjesec.
Ukno je bilo muotno. -> Prozor je bio mutan.
Niš ne vidiš -> Ništa ne vidiš
0. Moj taj je nesta.
1. Ni se ciepilo.
2. Je pljuknu žvakaćo.
```

Ponovno, kao što je spomenuto u 4.2., izazovno je bilo pravilno izdvojiti prijevode iz generiranih odgovora. Za model su ponovno korištene zadane postavke. Rezultati su prikazani u tablicama 4.8. i 4.9.

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	59.17	0.3465	38.03	0.4297
COPA	57.99	0.3511	31.97	0.5174
Mići Princ	57.70	0.3830	38.69	0.4327
Pjesme	66.83	0.2936	49.66	0.3009

Tablica 4.8. Rezultati GPT-4.1 s "zero-shot" upitom

Skup	chrF	CER	osnovni chrF	osnovni CER
Test	59.58	0.3294	38.03	0.4297
COPA	59.62	0.3200	31.97	0.5174
Mići Princ	58.46	0.3469	38.69	0.4327
Pjesme	62.88	0.3179	49.66	0.3009

Tablica 4.9. Rezultati GPT-4.1 s "3-shot" upitom

4.7. Usporedba rezultata

Iako rezultati nisu na visokoj razini, u njima se daju primijetiti određene pravilnosti. Kao prvo, svi su modeli postigli najbolje rezultate na skupu Pjesme, što je očekivano s obzirom na najvišu sličnost između dijalekta i standarda u tom skupu. Druga stvar koja se može primijetiti je da je, bez iznimke, najveći napredak u odnosu na osnovni model postignut na skupu podataka DIALECT-COPA , čemu opet pogoduje najniža polazna točka u osnovnom modelu. Unatoč tome, kod modela GPT-4.1 dolazi do boljih rezultata na skupu DIALECT-COPA nego na prijevodu Malog Princa, isto kao i u CER metrici na

modelu mT5. Konačno, može se primijetiti da modeli bez iznimke imaju bolje performance kada se skupu za učenje dodaju augmentirani primjeri, što je također očekivani za ovako mali skup podataka. Što se tiče usporedbe samih modela, GPT-4.1 je očekivano pokazao najbolje performanse, dok je mBART-50-large bio nešto iza. mT5-small pokazao se najslabijim odabirom za ovaj zadatak. Pokazuje se visoka korelacija između veličine modela i njegovih performansi.

5. Rasprava i zaključak

Nakon provođenja eksperimenata, glavni dojam koji se može steći je da eksperimenti nisu zadovoljili očekivanja. Rezultati koji su prikazani nisu na razini koja bi se zahtijevala od modela koji bi trebao standardizirati dijalektne tekstove na standardni jezik za daljnju obradu. Glavni razlog tomu leži u podacima, i to kroz dvije dimenzije: količinu i kvalitetu.

Količina podataka

Moguće je da bi za neki drugi zadatak obrade prirodnog jezika, poput klasifikacije ili označavanja, pa čak i takav zadatak vezan uz dijalekte (detekcija rečenica ili riječi u dijalektnom obliku) ova količina podataka bila dovoljna za neke prihvatljive rezultate. Međutim, čini se da zadatak standardizacije iziskuje nešto veći korpus podataka. U prilog tome ide i činjenica da su svi modeli pokazali bolje performanse po svim kriterijima kada su učeni na skupu podataka proširenim augmentiranim primjerima. Ti su augmentirani primjeri bili upitne kvalitete, ali svejedno su doprinijeli generalnoj sposobnosti modela da nauči generirati smislenu rečenicu na izlazu. Ostaje otvoreno pitanje koliko bi dobro modeli mogli prevoditi čakavsko narječje kada bi im se pružila odgovarajuća količina primjera.

Kvaliteta podataka

Obzirom na to iz kojih su izvora podaci prikupljani, očekivano je da će u njima biti određenih nekonzistentnosti. Čakavski rječnik tu je možda najkonzistentniji, obzirom da je stvaran s ciljem da primjeri sadrže doslovne prijevode. Iznenađujuće mnogo konzistentnosti prisutno je u zbirci pjesama, gdje se moglo očekivati i više različitosti s obzirom na pjesničku slobodu izražaja. Ta sloboda izražaja prisutna je u prijevodu Malog Princa, gdje se neke rečenice potpuno razlikuju po strukturi i redoslijedu riječi, iako zadržavaju

suštinu značenja. Slično je i sa skupom DIALECT-COPA . Treba uzeti u obzir da ti skupovi nisu stvarani da bi se koristili kao prijevod izvornog teksta na čakavski, već kao zasebne cjeline. Tu se ne dovodi u pitanje kvalitetu zasebnih rečenica na čakavskom narječju, već njihovu upotrebljivost za ovaj specifični zadatak kada ih se upari s izvornim tekstom. Međutim, kvaliteta u oba smisla upitna je kod augmentiranih primjera, budući da ni sam GPT-4.1 model nije pokazao visoke performanse na zadatku standardizacije.

Dojam je da bi jedan kvalitetan projekt stvaranja skupa podataka uz izvorne govornike i lingviste napravio mnogo po pitanju unaprjeđenja daljnjih eksperimenata u ovom uskom području.

Literatura

[1] O. Kuparinen, A. Miletić, i Y. Scherrer, "Dialect-to-standard normalization: A large-scale multilingual evaluation", u *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, i K. Bali, Ur. Singapore: Association for Computational Linguistics, prosinac 2023., str. 13 814–13 828. https://doi.org/10.18653/v1/2023.findings-emnlp.923

[2] G. Tang, F. Cap, E. Pettersson, i J. Nivre, "An evaluation of neural machine translation models on historical spelling normalization", u *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, i P. Isabelle, Ur. Santa Fe, New Mexico, USA: Association for Computational Linguistics, kolovoz 2018., str. 1320–1331. [Mrežno]. Adresa: https://aclanthology.org/C18-1112/

[3] R. van der Goot, A. Ramponi, A. Zubiaga, B. Plank, B. Muller, I. San Vicente Roncal, N. Ljubešić, Ö. Çetinoğlu, R. Mahendra, T. Çolakoğlu, T. Baldwin, T. Caselli, i W. Sidorenko, "MultiLexNorm: A shared task on multilingual lexical normalization", u *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, W. Xu, A. Ritter, T. Baldwin, i A. Rahimi, Ur. Online: Association for Computational Linguistics, studeni 2021., str. 493–509. https://doi.org/10.18653/v1/2021.wnut-1.55

[4] J. Richards i R. Schmidt, *Longman Dictionary of Language Teaching and Applied Linguistics*, 11 2013. https://doi.org/10.4324/9781315833835

[5] A. Vujić, *Opća i nacionalna enciklopedija*. Pro Leksis, Večernji list, 2007.

[6] R. Matasović, *Poredbneopovijesna gramatika hrvatskog jezika*. Zagreb: Matica hrvatska, 2008.

[7] N. Ljubešić, N. Galant, S. Benčina, J. Čibej, S. Milosavljević, P. Rupnik, i T. Kuzman, "DIALECT-COPA: Extending the standard translations of the COPA causal commonsense reasoning dataset to South Slavic dialects", u *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, Y. Scherrer, T. Jauhiainen, N. Ljubešić, M. Zampieri, P. Nakov, i J. Tiedemann, Ur. Mexico City, Mexico: Association for Computational Linguistics, lipanj 2024., str. 89–98. https://doi.org/10.18653/v1/2024.vardial-1.7

[8] L. Rijeka. (2024) Čakavski rječnik. [Mrežno]. Adresa: https://lokalpatrioti-rijeka.com/cakavski-rjecnik/

[9] M. Roemmele, C. A. Bejan, i A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning", u *2011 AAAI Spring Symposium Series*, 2011. [Mrežno]. Adresa: https://people.ict.usc.edu/~gordon/publications/AAAI-SPRING11A.PDF

[10] N. Ljubešić i D. Lauc, "BERTić - the transformer language model for Bosnian, Croatian, Montenegrin and Serbian", u *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, B. Babych, O. Kanishcheva, P. Nakov, J. Piskorski, L. Pivovarova, V. Starko, J. Steinberger, R. Yangarber, M. Marcińczuk, S. Pollak, P. Přibáň, i M. Robnik-Šikonja, Ur. Kiyv, Ukraine: Association for Computational Linguistics, travanj 2021., str. 37–42. [Mrežno]. Adresa: https://aclanthology.org/2021.bsnlp-1.5/

[11] N. Ljubešić, P. Rupnik, i T. Perinčić, "The "mići princ" text and speech dataset of chakavian micro-dialects", 2024., slovenian language resource repository CLARIN.SI. [Mrežno]. Adresa: http://hdl.handle.net/11356/1765

[12] A. de Saint-Exupéry, *Le Petit Prince*. Gallimard, 1943.

[13] ——, *Mali princ*. Mladost.

[14] R. Socher, Y. Bengio, i C. D. Manning, "Deep learning for NLP (without

magic)", u *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, M. Strube, Ur. Jeju Island, Korea: Association for Computational Linguistics, srpanj 2012., str. 5. [Mrežno]. Adresa: https://aclanthology.org/P12-4005/

[15] T. Mikolov, K. Chen, G. Corrado, i J. Dean, "Efficient estimation of word representations in vector space", 2013. [Mrežno]. Adresa: https://arxiv.org/abs/1301.3781

[16] F. Gers, J. Schmidhuber, i F. Cummins, "Learning to forget: continual prediction with lstm", u *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, sv. 2, 1999., str. 850–855 vol.2. https://doi.org/10.1049/cp:19991218

[17] D. Bahdanau, K. Cho, i Y. Bengio, "Neural machine translation by jointly learning to align and translate", 2016. [Mrežno]. Adresa: https://arxiv.org/abs/1409.0473

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, i I. Polosukhin, "Attention is all you need", 2023. [Mrežno]. Adresa: https://arxiv.org/abs/1706.03762

[19] J. Devlin, M.-W. Chang, K. Lee, i K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", 2019. [Mrežno]. Adresa: https://arxiv.org/abs/1810.04805

[20] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, i L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", u *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, i J. Tetreault, Ur. Online: Association for Computational Linguistics, srpanj 2020., str. 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[21] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, i L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation", 2020. [Mrežno]. Adresa: https://arxiv.org/abs/2001.08210

[22] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, i A. Fan, "Multilingual translation with extensible multilingual pretraining and finetuning", 2020.

[23] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, i P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", 2023. [Mrežno]. Adresa: https://arxiv.org/abs/1910.10683

[24] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, i C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer", 2021. [Mrežno]. Adresa: https://arxiv.org/abs/2010.11934

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, i P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer", *arXiv e-prints*, 2019.

[26] A. Radford i K. Narasimhan, "Improving language understanding by generative pre-training", 2018. [Mrežno]. Adresa: https://api.semanticscholar.org/CorpusID:49313245

[27] OpenAI, J. Achiam, S. Adler, i S. A. et al., "Gpt-4 technical report", 2024. [Mrežno]. Adresa: https://arxiv.org/abs/2303.08774

[28] J. Howarth, "Number of parameters in gpt-4 (latest data)", Exploding Topics blog, lipanj 2025., last updated June 17, 2025. [Mrežno]. Adresa: https://explodingtopics.com/blog/gpt-parameters

[29] A. Krizhevsky, I. Sutskever, i G. E. Hinton, "Imagenet classification with deep convolutional neural networks", u *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, i K. Weinberger, Ur., sv. 25. Curran Associates, Inc., 2012. [Mrežno]. Adresa: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[30] A. Møller, J. Dalsgaard, A. Pera, i L. M. Aiello, "Is a prompt and a few samples all you need? using gpt-4 for data augmentation in low-resource classification tasks", 04 2023. https://doi.org/10.48550/arXiv.2304.13861

[31] A. G. Møller, J. A. Dalsgaard, A. Pera, i L. M. Aiello, “The parrot dilemma: Human-labeled vs. llm-augmented data in classification tasks”, 2024. [Mrežno]. Adresa: https://arxiv.org/abs/2304.13861

[32] K. Papineni, S. Roukos, T. Ward, i W. J. Zhu, “Bleu: a method for automatic evaluation of machine translation”, 10 2002. https://doi.org/10.3115/1073083.1073135

[33] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries”, 01 2004., str. 10.

[34] S. Chauhan, P. Daniel, A. Mishra, i A. K. and, “Adableu: A modified bleu score for morphologically rich languages”, *IETE Journal of Research*, sv. 69, br. 8, str. 5112–5123, 2023. https://doi.org/10.1080/03772063.2021.1962745

[35] “Kaggle”, accessed: 2025-04-15. [Mrežno]. Adresa: https://www.kaggle.com/

[36] “Nvidia t4 tensor core gpu”. [Mrežno]. Adresa: https://www.nvidia.com/en-us/data-center/tesla-t4/

[37] “mt5 small”, accessed: 2025-04-15. [Mrežno]. Adresa: https://huggingface.co/google/mt5-small

[38] “Openai api”, accessed: 2025-04-15. [Mrežno]. Adresa: https://platform.openai.com/docs/api-reference

Sažetak

Model i skup podataka za prevođenje čakavskog narječja na standardni hrvatski jezik

Florijan Sandalj

U ovom radu istražuje se standardizacija čakavskog narječja na standardni hrvatski jezik korištenjem metoda obrade prirodnog jezika. Opisuju se izvori podataka te njihova priprema i konstrukcija konačnog skupa podataka. Za proširenje skupa za učenje koristi se augmentacija podataka velikim jezičnim modelom GPT-4.1 Opisuje se teorijska pozadina potrebna za razumijevanje korištenih arhitektura. Na zadatku standardizacije testiraju se četiri arhitekture: BERTić + LSTM, mT5-small, mBART-50-large i GPT-4.1. Prikazuju se i komentiraju rezultati.

Ključne riječi: obrada prirodnog jezika; čakavsko narječje; standardizacija dijalekta; augmentacija podataka

Abstract

Model and dataset for translating the Chakavian dialect into the standard Croatian language

Florijan Sandalj

This paper explores the standardization of the Čakavian dialect into the standard Croatian language using natural language processing (NLP) methods. It describes the data sources, their preparation, and the construction of the final dataset. To expand the training set, data augmentation is performed using the large language model GPT-4.1. The theoretical background necessary for understanding the architectures used is presented. Four architectures are tested on the standardization task: BERTić + LSTM, mT5-small, mBART-50-large, and GPT-4.1. The results are presented and discussed.

Keywords: natural language processing; chakavian dialect; dialect standardization; data augmentation