UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Jovan Marković

FINO PODEŠAVANJE JEZIČKOG MODELA BERT ZA ANALIZU SENTIMENTA

master rad

Beograd, 2025.

Mentor:

prof Aleksandar KARTELJ, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

prof Mladen NIKOLIĆ, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

prof Vladimir FILIPOVIĆ, redovan profesor
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Svim ljudima koje sam upoznao tokom studiranja na fakultetu

Naslov master rada: Fino podešavanje jezičkog modela BERT za analizu sentimenta

Rezime: Ovaj rad predstavlja različite načine za fino podešavanje velikih jezičkih modela. U radu se analiziraju koncepti velikih jezičkih modela, njihova struktura, osnovne karakteristike i načini na koji se njihova generativna priroda može prilagoditi potrebama za rešavanjem specifičnih zadataka. Kao centralna tema, vrši se analiza primene različitih tehnika finog prilagođavanja modela BERT za rešavanje problema analize sentimenta. Cilj rada je upoređivanje potrebnih resursa i poboljšanja generisanih odgovora prilikom realizacije svake od obrađenih tehnika.

Ključne reči: veliki jezički modeli, BERT, analiza sentimenta, fino prilagođavanje

Sadržaj

Glava 1

Uvod

Veliki jezički modeli (eng. Large Language Models - LLMs) predstavljaju modele istrenirane nad velikim skupom tekstualnih podataka. Oni sadrže milijarde parametara čiji je zadatak da procesiraju i razumeju tekst napisan na prirodnom jeziku, kao i da pruže adekvatan odgovor, najčešće ponovo u vidu teksta prirodnog jezika. Jedna od značajnih pogodnosti kojima raspolažu je da za njihovo treniranje nisu potrebni labelizovani podaci niti je potrebna velika pažnja prilikom preprocesiranja ulaznih podataka, već su u stanju da samostalno izvlače obrasce koji omogućavaju adekvatno shvatanje konteksta.

Obrada prirodnih jezika (eng. Natural Language Processing - NLP) predstavlja oblast koja se bavi razumevanjem konteksta i obradom prirodnih jezika unutar šire oblasti veštačke inteligencije (eng. Artificial Intelligence - AI) . Iako je postojala i ranije, najistaknutije u polju prevođenja teksta na različitim jezicima, obrada prirodnih jezika dobija na značaju i popularnosti tek u skorije vreme, kada se javljaju veliki jezički modeli modeli poput ChatGPT-ja kompanije OpenAI [1], a kasnije i modela LLaMA kompanije Meta [2], prilagođeni za ostvarivanje njenih ciljeva. Ono što je doprinelo njihovom naglom uspehu je mogućnost obavljanja velikog broja raznorodnih zadataka, kao i jednostavnost korišćenja. Za njihovo korišćenje nije potrebno posebno predzanje, već je zadatak razumevanja zahteva, njihova celokupna obrada i generisanje odgovarajućih izlaza prebačena na stranu modela. Na ovaj način, korisnicima je omogućeno da na intuitivan način zadaju upite i da očekuju jednako intuitivne rezultate koje model pruža kao odgovor.

Cena pravljenja sopstvenih jezičkih modela može biti značajna zahtevajući dragocene resurse [3]. Zbog toga je često praktično izabrati i iskoristiti već postojeće, istrenirane modele i prilagoditi ih sopstvenim potrebama. Popularnost jezičkih mo-

dela ogleda se i u njihovoj širokoj dostupnosti na internetu, kao i konstantnim novim modelima koji se mogu koristiti. Razvijaju se i posebne platforme koje omogućavaju preuzimanje modela, kao što je Hugging Face [4].

Prema vrsti zadataka koje mogu da obavljaju, neki modeli prebacuju fokus na obradu ulaznog teksta, neki na generisanje izlaznog teksta, dok je nekim modelima podjednako bitan ulaz kao i izlaz. Ovo je ostvareno korišćenjem posebnih gradivnih blokova, enkodera (eng. Encoder) i dekodera (eng. Decoder). Enkoder je modul zadužen za shvatanje značenja ulaznog teksta, dok je dekoder zadužen za generisanje izlaznog teksta. Moguća je i implementacija enkodera i dekodera u istom modelu [**?**]. Na ovaj način možemo da razlikujemo tri vrste arhitekture modela:

1. Enkoder arhitektura (eng. Encoder architecture)
   predstavnik: BERT [5]

2. Dekoder arhitektura (eng. Decoder architecture)
   predstavnik: GPT

3. Enkoder-dekoder arhitektura (eng. Encoder-Decoder architecture)
   predstavnik: T5 [6]

Razlike u arhitekturi modela impliciraju da vrsta problema koji se rešava određuje i vrstu modela koji se treba koristiti za njegovo rešavanje. Na primer, za rešavanje problema klasifikacije teksta ili analizu sentimenta potrebno je koristiti modele zasnovane na enkoder arhitekturi, za generaciju teksta ili kreativno pisanje modele zasnovane na dekoder arhitekturi, dok je za probleme prevođenja ili sumiranje najbolje koristiti modele zasnovane na enkoder-dekoder arhitekturi.

Kako su unapred trenirani i prilagođeni širokom kontekstu u svrhu razumevanja velike količine različitih podataka, oni predstavljaju modele sa širokim generativnim znanjem. Ova činjenica implicira mogućnost prilagođavanja modela za konkretne potrebe, u svrhu dobijanja preciznijih i pouzdanijih odgovora. U tom cilju javljaju se različite metode, koje se nazivaju finim podešavanjem (eng. fine-tuning) jezičkih modela. U suštini, fino podešavanje predstavlja treniranje celokupnih ili samo odabranih delova modela, u svrhu dobijanja preciznije generisanih izlaza na osnovu priloženih ulaznih instanci.

Zbog svoje kompleksne i memorijski zahtevne strukture, promene na modelima mogu oduzeti dragoceno vreme i potrebne resurse ukoliko ih želimo prilagoditi vlastitim potrebama. Ovo je razlog zašto je potrebno istražiti i uporediti različite

metode finog prilagođavanja, kako bismo znali koji metod je potrebno primeniti u zavisnosti od dostupnih resursa i potrebnih performansi. U ovom radu biće prikazane sledeće metode finog podešavanja modela:

1. potpuno fino podešavanje (eng. full fine-tuning)
2. fino podešavanje glave za klasifikaciju (eng. classification head fine-tuning) [7]
3. fino podešavanje zasnovano na adapterima (eng. Adapter-based fine-tuning) [8]
4. fino podešavanje zasnovano na matricama niskog ranga (eng. Low Rank Adaptation - LoRA) [9]

Prvi pristup utiče na celokupnu strukturu modela, dok su ostali pristupi vrste pristupa zasnovanog na parametarski efikasnom finom podešavanju (eng. Parameter-Efficient Fine-Tuning - PEFT) [10], koji za cilj ima brže i manje zahtevno prilagođavanje modela, koje ne utiče na celokupnu strukturu, već za odabrani podskup parametara.

Iako se tehnike finog podešavanja koje će biti izložene u ovom radu mogu primeniti na sve vrste modela fokus će biti prebačen na Google-ov model BERT, specijalizovan za razumevanje ulaznih podataka, primenjen na poznat problem analize sentimenta [11]. Kao konkretan primer, biće iskorišćen skup recenzija za sajt IMDB [12] koji se često koristi za potrebe testiranja performansi različitih jezičkih modela.

Glava 2

Osnovni koncepti

2.1 Obrada prirodnih jezika

Obrada prirodnih jezika (eng. Natural Language Processing - NLP) je široka oblast koja je počela da se razvija davno pre pojave velikih jezičkih modela, već oko 30-ih godina prošlog veka [13]. U svojoj suštini odnosi se na mašinsko razumevanje teksta napisanog na prirodnom jeziku. Neke od oblasti rada koje ovaj pojam obuhvata su prepoznavanje govora, prevođenje tekstova, sumarizacija teksta itd. Primeri popularnih aplikacija čija suština leži u obradi prirodnih jezika su virtualni asistenti (Alexa, Cortana, Siri...), internet pretraživači, prevodioci itd.

Tekst koji se obrađuje može biti struktuiran i nestruktuiran. Primeri nestruktuiranog teksta:

1. „Milica je pošla u školu."
2. „Kupi jaja i mleko."

Primeri struktuiranog teksta:

subjekat: Milica,
objekat: škola,
radnja: poći

ili

<namirnice>
<predmet>jaja</predmet>
<predmet>mleko</predmet>
</namirnice>

Struktuiran tekst je lakši za obradu i razumevanje mašinama, međutim ljudima nije prirodno da komuniciraju na ovaj način. Sa druge strane, nestruktuiran tekst je način na koji ljudi komuniciraju, ali je mašinama teško da shvate njegov kontekst i strukturu.

Proces prevođenja nestruktuiranog u struktuiran tekst naziva se shvatanje prirodnih jezika (eng. Natural Language Understanding - NLU) [14], dok se u suptornom smeru primenjuje generisanje prirodnih jezika (eng. Natural Language Generation) [15].

2.2 Tokenizatori

Često se misli da je jezički modeli rade sa sirovim tekstovima podataka. Ovo, međutim, nije tačno. Reči predstavljaju niske slova, odnosno simbola, sa kojima je mašinama teže da rade nego sa brojevima. Zbog toga se unutar modela vrše transformacije rečenica u liste korespodentnih numeričkih vektora kako bi mašinama bio olakšan rad.

Transformacija ulaznih podataka u vektore koji se prosleđuju modelima naziva se tokenizacija [16], a obavlja se pomoću posebnih komponenti koje se nazivaju tokenizatori. Tokenizatori preuzimaju ulazne podatke, vrše njihovu obradu, a na izlaz dostavljaju određen skup tokena. Tokeni mogu biti reči ili delovi reči koji su kodirani na određen način, a zatim su prosleđeni odgovarajućim jezičkim modelima za potrebe treniranja.

Različiti tokenizatori kodiraju skup reči na različite načine. Prilikom početnog treniranja modela nad ulaznim podacima, način rada tokenizatora može biti značajan za krajnje mogućnosti modela [17] [18]. Zbog različitog načina kodiranja, prilikom rada sa nekim unapred istreniranim jezičkim modelima, bitno je primeniti isti tokenizator, jer u suprotnom može doći do značajnog umanjenja moći modela.

Kako su tokenizatori programi koji prepoznaju obrasce u tekstu i rečima i dodeljuju im jedinstvene vrednosti na osnovu skupa na kojem su trenirani, moguće je da se desi i da za prosleđen simbol ili reč ne postoji odgovarajući token koji se može generisati. U ovom slučaju tokenizatori vraćaju specijalan token koji se naziva nepoznat token (eng. unknown token). Sa svakim neprepoznatim tokenom, gubi se deo informacija, ali ukoliko se ne javlja u velikoj meri, nije potrebno posebno rukovanje u ovim slučajevima.

Pored transformisanja rečenica u niz tokena, tokenizatori takođe imaju i ulogu konvertovanja skupa tokena nazad u rečenice čitljive krajnjim korisnicima.

// Opciono dodaj primer tokenizatora

2.3 Transformatori

Od objavljivanja dokumenta Attention is all you need [19], koji je sastavio tim koji je radio u Google-u, jezički modeli se znatno poboljšavaju i prethodno iscrpno korišćenje rekurentnih mreža i duge kratkoročne memorije u svrhu razumevanja i obrade prirodnih jezika zamenjuje se Transformatorima (eng. Transformers).

Kako se u dokumentu navodi, Transformatori su sastavljeni iz tri dela koji se nazivaju matrice pažnje (eng. Attention Matrix), neuronske mreže sa propagacijom unapred (eng. Feedforward Neural Network - FNN) kao i normalizacioni sloj (eng. Layer Normalization). Naizmeničnim smenjivanjem ovih slojeva, omogućeno je efikasno izvršavanje zadataka stavljenih pred modele. Glavna prednost Transformatora je u tome što omogućavaju veliku paralelizaciju izračunavanja korišćenjem velikog broja matrica.

Matrice pažnje

Osnovna uloga ovog sloja je prilagođavanje svake reči određenom kontekstu u kojem se javlja. Ukoliko, na primer, imamo reč jezik, od tokenizatora dobijamo odgovarajuću vektorsku reprezentaciju te reči, ali se ona menja u sloju matrica pažnje u zavisnosti od toga na kakav se jezik odnosi (programski jezik, organ u telu, prirodni jezik...).

Kako bi se odredio kontekst svake reči u tekstu, neophodno je pratiti sve reči koje joj prethode kako bi se njeno značenje približnije odredilo [1]. Broj reči koji se prati naziva se veličina konteksta modela. Poželjno je da veličina konteksta bude velika kako se informacije ne bi gubile u dužim konverzacijama, međutim ona direktno utiče na veličinu matrica pažnje koje su jednake njenom kvadratu.

Način na koji je predstavljen ovaj sloj opisuje se formulom:

[1] Bilo bi moguće implementirati rad matrica pažnje tako da na svaku reč utiču reči i ispred i iza svake reči, ali na taj način bi se izgubio uzročno-posledični odnos i model ne bi postizao dobre rezultate

$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$[2]

Ova transformacija predstavlja samo jedan od mnogo slojeva koji se paralelno računaju kako bi predložili izmenu u vektoru koji određuje svaku reč. Kada se svi oni sumiraju i dodaju na originalan vektor, dobija se konačna nova vrednost tog vektora.

Neuronske mreže sa propagacijom unapred

Neuronske mreže sa propagacijom unapred su jedna od ključnih arhitektura u oblasti mašinskog učenja kao i obrade prirodnih jezika[20]. One predstavljaju jedan od osnovnih vidova neuronskih mreža u kojima se informacije prenose u jednom smeru. U slučaju transformatora, ovaj sloj služi kao mesto na kojem se rezonuje, uči i ubacuje znanje koje model poseduje.

Neuronske mreže se primenjuju međusobno nezavisno nad svakim tokenom iz matrice koju generišu matrice pažnje. Ovo omogućava visoku paralelizaciju procesa.

Uopšteno, tokom procesa transformacije pojedinačnih tokena iz ulaznih do izlaznih, prolazi se kroz više slojeva. Najpre se primenjuje linearna funkcija koja ulazne podatke prebacuje u visokodimenzionalni prostor u kojem primenjuje nelinearnu funkciju (npr. ReLU), da bi na kraju izvođenjem novog matričnog množenja vratilo konačni vektor na dimenzije početnog. Prebacivanje u visokodimenzioni prostor je bitan aspekt u modelovanju neuronskih mreža koji omogućuje modelu veći opseg parametara koje koristi (a samim tim i veće izvođenje zaključaka). U originalnom radu o transformatorima, predstavljena je formula koja opisuje ovaj proces:

$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$

[2]Ovime je predstavljena matrica veličine |veličina konteksta| x |veličina konteksta| gde su po kolonama smeštene vrednosti Q (eng. Query), koje predstavljaju rezultat matričnog množenja posebnih podesivih matrica M_Q(eng. Query Matrix) sa svakom reči u kontekstu, dok su vrednosti K (eng. Key) po redovima nastali množenjem sa drugom matricom M_K (eng. Key Matrix). Vrednosti u matrici pažnje su skalarni proizvodi koji su blizu 1 ukoliko postoji korespodencija između dve reči. Zbog toga što želimo da na reč utiču samo reči koje joj prethode, ova matrica je gornje trougaona. Po svakoj koloni se primenjuje softmax funkcija kako bi se dobile verovatnoće koje konfigurišu u tekstu. Promenljiva T predstavlja temperaturu i obično je u opsegu od 1 do 2, $\sqrt{d_K}$ pomaže za potrebe stabilnosti prilikom računanja, a predstavlja koren od broja dimenzija vektora K. Svaki od vektora V dobija se kao proizvod posebne prilagodljive matrice M_V sa vektorima koji predstavljaju reprezentaciju određene reči i u apstraktnom smislu predstavlja vektor koji treba dodati na ostale reči ukoliko želimo da objedinimo njegova svojstva da budu primetna. Nakon primene softmax funkcije, po kolonama se radi još jedan skalarni proizvod sa vektorima V, kako bi se primenio proporcionalni uticaj svake od prethodnih reči. U literaturi, matrica M_V često je prikazana kao proizvod dve matrice, prva koja preslikava vektor u prostor koji odgovara veličini vektora K i Q, i drugu koja to preslikavanje vraća u originalnu dimenziju. Ova transformacija naziva se i transformacija niskog nivoa (eng. low-rank transformation)

U ovoj formuli W_1 i W_2 predstavljaju prilagodljive matrice, dok su b_1 i b_2 vektori slobodnih članova tih matrica, dok je funkcijom max() predstavljena ReLU funkcija.

Normalizacioni sloj

Normalizacioni slojevi su komponente koje u transformatorima služe za stabilizaciju i ubrzavanje procesa treniranja. Oni se postavljaju između slojeva pažnje i slojeva neuronskih mreža. Ovim slojevima se rešavaju problemi nestajućih i eksplodirajućih koeficijenata, a doprinose računanju gradijenta prilikom propagacije.

U Vaswani-jevom radu, izlaz ovog sloja se opisuje formulom:

$LayerNorm(x + Sublayer(x))$

Ovde Sublayer() predstavlja funkciju koja vrši transformacije na vektorom, u sloju matrica pažnje ili neuronskih mreža

Ovaj pristup se naziva Post-LayerNorm i u njemu se proces normalizacije izvršava nakon sabiranja rezultata sa početnim vektorom. Nasuprot ovoga, postoji i Pre-LayerNorm pristup koji se javlja u novije vreme a u kojem se normalizacija vrši pre početka obrade podataka.

$X + Sublayer(LayerNorm(X))$

Razlika između ova dva pristupa je u tome da li će se normalizacija podataka pre ili nakon obrade slojeva u transformatoru. Postoje radovi koji upoređuju ova dva pristupa[21].

Svi predloženi slojevi međusobno se smenjuju i u svakoj iteraciji približavaju cilju. Matrice pažnje služe za međusobno shvatanje odnosa između susednih reči i time njihovo prilagođavanje kontekstu u kojem se javljaju. Nasuprot tome, prolazak tokena kroz neuronsku mrežu obavlja se samostalno i nezavisno od okoline. Normalizacioni slojevi su postavljeni između njih kao konfiguracioni slojevi koji osiguravaju stabilna i brza izračunavanja. U svakom transformatoru ova tri sloja se međusobno smenjuju i ponavljaju nekoliko desetina ili stotina puta kako bi se na kraju dobilo zadovoljavajuće rešenje.

2.4 BERT model

Model BERT[5] nastao je 2018. godine od strane stručnjaka iz Guglovog tima. Motivacija za razvoj ovog modela bilo je zapažanje da su dotadašnji modeli koristili

isključivo čitanje konteksta sa leve ili sa desne strane. Ovaj model, čije puno ime stavlja akcenat upravo na njegovu karakteristiku da pri analizi rečenica koristi kontekst i sa leve i sa desne strane (eng BERT - Bidirectional Encoder Representations from Transformators), pomerio je granice mogućnosti rezonovanja modela.

Unidirekcioni modeli (oni koji koriste samo kontekst sa leve ili sa desne strane svake reči) pokazuju se dobro na zadacima generisanja teksta, međutim često se javljaju sub-optimalna rešenja usled toga što se javlja ograničenje u količini konteksta koji definiše svaku reč. Sa druge strane, za potrebe razumevanja celokupnog konteksta, bidirekcioni modeli se pokazuju kao bolji izbor. U ove probleme spadaju klasifikacija teksta, odgovaranje na pitanja ili analiza sentimenta.

Bidirekciona priroda BERT modela omogućava jedinstven način treniranja modela, optimizovanjem MLM (Masked Language Model) funkcije cilja. Ovo se postiže tako što MLM nasumično maskira tokene sa ulaza, a cilj je da se pomoću konteksta uspešno izvrši predikcija maskiranih tokena. Istovremenim spajanjem konteksta sa obe strane, ovaj model naziva se i dubokim bidirekcionim modelom, nasuprot plitkih bidirekcionih modela, nalik na ELMo[22] model koji koristi konkatenaciju dva odvojena unidirekciona sloja.

DistilBERT model

Iako je BERT model postigao značajne rezultate, njegov nedostatak je veličina i složenost samog modela. Za potrebe nekih zadataka mogu se priuštiti malo lošije performanse zarad skladištenja manjeg modela i povećanja brzine obrade zahteva. Za takve potrebe je 2019. godine predstavljen DistilBERT[23] model. Kao što i sugestira njegovo ime, ovaj model dobijen je od originalnog procesom destilacije znanja.

Destilacija znanja(eng. Knowledge distillation) [24] je tehnika kompresije modela u kojoj se manji model - student (eng. student model) trenira da reprodukuje ponašanje većeg, kompleksnijeg modela - učitelja (eng. teacher model). Model učitelj je tipično istreniran da daje dobre rezultate za specifičan zadatak. Model student uči na osnovu njega tako što se trudi da minimizuje svoje izlaze sa izlazima generisanim od učitelja, često koristeći meke labele, gde su vrednosti za svaku u rasponu [0,1], umesto tvrdih labela (vrednosti strogo 0, 1). Ovaj proces omogućava manjem i bržem studentu da dostigne znanje i sposobnost generalizacije kompleksnijeg modela.

Prilikom pravljenja destilovane verzije osnovnog BERT modela, najznačajnija promena je smanjenje broja slojeva za faktor 2, tako da DistilBERT ima samo 6

Model	IMDb (Accuracy)	SQuAD (EM/F1)
BERT	93.46	81.2 / 88.5
DistilBERT	92.82	77.7 / 85.8

Tabela 2.1: Upoređivanje na zadacima: IMDB(test accuracy) i SQuAD (EM/F1 mera na validacionom skupu)

Model	#param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT	110	668
DistilBERT	66	410

Tabela 2.2: Upoređivanje veličina modela i vreme potrebno za kompletan prolaz kroz GLUE STS-B zadatke na CPU gde je batch size 1

slojeva, u odnosu na 12, koliko ima BERT. Nakon sprovođenja eksperimenta, autori navode da je njihova verzija modela koristi za 40% manje parametara, a brzina izvršavanja 60% veća, a sve to zadržavajući 97% mogućnosti za rezonovanje originalnog modela. Kao testni primeri za proveru sposobnosti ovih modela, korišćen je zadatak klasifikacije nad skupovima IMDB, koji se koristi za analizu sentimenta, gde se testira tačnost modela, kao i skup SQuAD, koji predstavlja zadatak odgovaranja na pitanje, a testira EM/F1 metriku na validacionom skupu. Rezultati ovih eksperimenata dati su u tabeli 2.1. Kao pokazatelj vremena potrebnog za obradu zahteva, korišćen je GLUE STS-B zadatak (analiza sentimenta). U tabeli 2.2 u kolonama su predstavljeni broj parametara modela, kao i vreme obrade svih zahteva.

2.5 Problem analize sentimenta

Problem analize sentimenta (eng. Sentiment analysis) predstavlja problem klasifikacije teksta u kojima je cilj odrediti da li odražava pozitivno ili negativno mišljenje o nekoj temi. Zbog svoje jednostavnosti za shvatanje, velike količine skupova podataka koja se može pronaći iz različitih oblasti, kao i različitih mogućnosti za generisanje cilja, ovaj problem predstavlja jedan od najosnovnijih koji se koriste u svrhu testiranja rezonovanja jezičkih modela. Skupovi tekstova koji se mogu pronaći poreklo mogu da vode sa različitih mesta, kao što su internet forumi, ocene proizvoda poručenih preko interneta, komentari na društvenim mrežava, ocene filmova... Kao cilj klasifikacije česte su podele na dve (pozitivno ili negativno) ili tri (pozitivno, negativno ili neutralno) klase, a moguće je i na izlazu generisati brojčanu vrednost

(na primer ocena od 1 do 10, gde 1 predstavlja najlošiju ocenu, a 10 najbolju).

Modeli specijalizovani za rešavanje problema analize sentimenta svoju primenu mogu takođe naći na različitim mestima. Značajni mogu biti u sistemima preporuke gde se koriste informacije o utiscima proizvoda, ali takođe i na društvenim mrežama gde se prati koliku reputaciju imaju određeni brendovi ili organizacije. Omogućavaju praćenje uspešnosti filmova/serija, ali i političkih kampanja ili događaja u svetu. Takođe bitna stavka je i korišćenje za potrebe ispitivanja finansijskih tržišta, gde je bitno prepoznati mogući uspeh nekog proizvoda u ranoj dobi. Pored nabrojanih, postoje i razne druge oblasti [25].

Zbog svoje raznovrsnosti, kako granulacije na kojoj se analiza može vršiti (analiza dokumenata, pasusa, rečenica, komentara...)[26], tako i kompleksnosti samih tekstova (analiza emoji-ja, lokalizama, formalan ili neformalan govor, reči u kojima su sva slova velika, sarkazma u tekstu...) u ovim problemima mogu se javljati razni izazovi koje modeli treba uspešno da prevaziđu [27].

Analiza sentimenta jedna je od starijih problema u oblasti klasifikacije, i za njeno rešavanje mogu se koristiti mnoge metode, kao što su klasične metode nalik na Naivni Bajes[28] ili sistem potpornih vektora[29], metode dubokog učenja nalik na konvolutivne[30] ili rekurentne neuronske mreže[31], LSTM (Long-short term memory)[32]... a moguće je i koristiti jezičke modele zasnovane na transformatorima, kao što su BERT, DistilBERT, RoBERTa...

Trenutno je dostupan veliki broj različitih skupova za probleme analize sentimenta, a neki od najpoznatijih su SST (Stanford Entiment Treebank)[33], Amazon products review [34], Sentiment140 [35], IMDBdataset [12] i drugi.

IMDB dataset

IMDB dataset je skup podataka nastao je koristeći informacije sa jednog od najpoznatijih sajtova za ocenjivanje filmova - IMDB (Internet Movie Database). Ovo je jedan od najkorišćenijih skupova za testiranje uspešnosti modela u problemima binarne klasifikacije sentimenta. Sastoji se od 50 000 instanci koje predstavljaju komentare filmova, ravnomerno podeljenih na pozitivne i negativne. Ovaj skup nastao je 2011. godine od strane Andrew L. Maas-a i od tada se ekstenzivno koristi.

Ovaj skup sastoji se od ravnomerno podeljenih instanci na trening i test bez preklapanja. Dodatno, ocene su takođe jednako podeljene, što omogućava u eliminaciji pristrasnosti određenim ciljnim klasama.

Glava 3

Fino podešavanje velikih jezičkih modela

Fino podešavanje nastaje prilagođavanjem pre-treniranog modela kao osnove na manjem domenski-određenom skupu. Ono što razlikuje ovu metodu od klasičnog treniranja modela, jeste to što se kao osnova koristi model koji već ima opsežno znanje i percepciju lingvističkih pojmova i njihovih odnosa. Zbog toga, fino podešavanje predstavlja dodatni korak koji može značajno doprineti rezultatima izvršavanja modela u oblastima za koje je dodatno treniran.

3.1 Značaj i uloga finog podešavanja

Fino podešavanje nadograđuje postojeće znanje modela, prilagođava ga oblasti u kojoj se očekuje njegovo korišćenje, poboljšavajući performanse dok istovremeno koristi manje kompjuterskih resursa ili podataka pri njihovoj obradi. Zbog svega ovoga, primena finog podešavanja postala je popularna u NLP zadacima, pogotovo u oblasti klasifikacije teksta, analize sentimenta i generisanja odgovora.

Pogodnosti korišćenja finog podešavanja, ukoliko je ono moguće, su brojne, a neke od njih su date u nastavku:

- **Prenos znanja (eng. Transfer Learning)** Fino podešavanje omogućava modelu da zadrži svoje stečeno generalno znanje i dalje ga prilagodi specifičnim zadacima. Ovaj proces značajno smanjuje vreme i resurse u odnosu na treniranje modela od starta, kako model već poseduje snažne osnove u shvatanju prirodnih jezika [36].

- **Smanjena količina potrebnih podataka** Kako pre-trenirani modeli već poseduju izvesnu količinu znanja, fino podešavanje je usmereno većinom ka shvatanju suptilnijih veza između relevantnih pojmova u definisanom domenu, nije potrebna velika količina podataka za njegovo prilagođavanje. Ovo je naročito korisno u specijalizovanim oblastima gde ne postoje veliki označeni skupovi ili u primeni na jezike za koje ne postoje ekstenzivni resursi[37].

- **Poboljšana generalizacija** Kako fino podešavanje obično obohvata prilagođavanje samo pojedinih elemenata modela, umesto svih elemenata što je slučaj kod inicijalnog treniranja, na ovaj način može se zadržati visok stepen generalizacije početnog modela, smanjujući mogućnost za preprilagođavanjem modela [38].

- **Efikasna isporuka modela** Fino podešeni modeli računski su efikasniji za primenu u aplikacijama u stvarnom svetu, kako ne zahtevaju pre-treniranje u punom obimu. Ovo ih čini podesnim za isporuku u produkcionim okruženjima gde resursi mogu biti ograničeni. Dodatno, fino podešavanje omogućava prilagođavanje modela bez potrebe za potpuno novom arhitekturom.

- **Prilagođavanje na različite zadatke** Fino podešeni modeli omogućavaju jednom pre-treniranom modelu da se prilagodi na širok spektar NLP zadataka. Umesto da postoji poseban model za svaki zadatak, fino podešenim modeli može se promeniti svrha bez ekstenzivnih modifikacija.

- **Performanse u domenskim zadacima** Fino podešavanje omogućava modelima visoke performanse u specijalizovanim oblastima, nalik na biomedicinu, pravo ili finansije, kao i na nove i nedovoljno dokumentovane oblasti. Ukoliko pre-treniran model posveti pažnju domenski specifičnim podacima, on može naučiti posebne žargone i pojmove i njihove odnose i kasnije ih primeniti u praksi[39].

- **Brža konvergencija** Kako je pre-treniran model već prilagođen generalnom shvatanju jezika, dodatne izmene konvergiraju brže nego kod modela koji sve moraju da preračunaju ispočetka. Ovo rezultuje manjem vremenu treniranja kao i manjim računskim zahtevima, što omogućava brže eksperimentisanje sa različitim tehnikama finog podešavanja i čini ga preferiranim metodom za mnoge primene NLP-a u stvarnom svetu.

3.2 Vrste finog podešavanja

Kako fino podešavanje predstavlja opšti način za usmeravanje modela ka specifičnim oblastima za koje će biti korišćen, njegovo izvršavanje može se ostvariti na različite načine. U opštem smislu, metode korišćene u svrhu finog podešavanje mogu se implementirati imajući u vidu 3 različita pristupa:

- **Po podacima koji se koriste**
- **Po broju datih primera**
- **Po strategiji odabira parametara koji se modifikuju**

1. Po podacima koji se koriste

U ovoj podeli vrši se fino podešavanje modela kroz različite pružene podatke a najčešće zavisi od problema koji se rešavaju.

- **Nadgledano fino podešavanje** – Model je treniran nad labelisanim skupom, kako bi dodatno prilagodio svoje parametre. Korisno za rešavanje problema klasifikacije ili mašinske prevođenja teksta.
- **Nenadgledano fino podešavanje** – Model se trenira na neoznačenim podacima u svrhu daljeg shvatanja veza između elemenata teksta. Korisno ukoliko model želimo da upoznamo sa usko specifičnim oblastima ili oblastima za koje nema puno podataka.
- **Fino podešavanje kroz prompt instrukcije** – Umesto modifikacije parametara modela, modelu se nagoveštava način na koji treba da generiše odgovor. Korisno ukoliko nam je struktura ili veličina generisanog odgovora bitna.

2. Po broju datih primera

U ovom slučaju kao značajna komponenta figuriše količina pruženih podataka. Ukoliko imamo ograničen skup podataka, i treniranje na svega nekoliko pruženih instanci može poboljšati performanse modela.

1. **Učenje bez primera (eng. Zero-shot learning)** – U ovom slučaju model se izvršava bez ikakvog predznanja o specifičnostima za zadatu oblast, oslanjajući se na svoje pre-trenirano znanje.

2. **Učenje sa malo primera (eng. Few-shot learning)** – U ovom slučaju modelu ja pružen mali skup označenih podataka (često svega 2-3 primera), najčešće u svrhu uspešne generalizacije problema.

3. **Puno fino podešavanje** – Model se trenira na velikom skupu kako bi se potpuno prilagodio postavljenim zadacima.

3. Po strategiji odabira parametara koji se modifikuju

U zavisnosti od odabrane strategije, kao i količine parametara kojima je omogućeno treniranje može se balansirati između prilagodljivosti modela specifičnim zadacima kao i zadržavanju generalizovanog znanja koje model poseduje. Ovo je oblast finog podešavanja koja poseduje najveće mogućnosti i mnogo različitih načina za ostvarivanje. Zbog toga će se u ovom radu najviše pažnje posvetiti upravo ovim metodama finog podešavanja i ispitati njihov uticaj na model.

- **Potpuno fino podešavanje (eng. Full Fine-Tuning)** – Svi parametri modela označavaju se kao mogući za treniranje. Ovaj pristup zahteva najviše računskog vremena kako utiče na celokupan model.

- **Parametarski efikasno fino podešavanje (eng. PEFT - Parameter-Efficient Fine-Tuning)** – Širok spektar različitih strategija za fino podešavanje, u opšem smislu odnose se na zamrzavanje velikog broja parametara modela, dok se za treniranje koristi manji broj postojećih ili dodatih parametara kako bi se ubrzao proces treniranja i prilagodio model bez gubitka moći generalizacije. U ovu kategoriju potpadaju mnogi pristupi, od kojih su najpoznatiji:

  - **Fino podešavanje zasnovano na adapterima (eng. Adapter-based fine-tuning)** – Modelu se svi parametri označavaju kao netrenirajući, a ubacuje se dodatni sloj adaptera koji se prilagođava. Na ovaj način smanjuju se računski zahtevi, a dosta prostora se ostavlja eksperimentisanju sa slojevima adaptera i njihovog uticaja na specifične probleme.
  - **Fino podešavanje zasnovano na matricama niskog ranga (eng. Low-Rank Adaptation)** – Popularan efikasan metod koji smanjuje korišćenje memorije trenirajući podskup težinskih matrica.

Neki pristupi finom podešavanju kombinuju različite nabrojane tehnike kako bi maksimalno iskoristili mogućnosti svojih modela. Odabir korišćene tehnike pre svega zavisi od problema koji se rešava, dostupnih kompjuterskih resursa i količine dostupnih podataka, kao i poznavanja unutrašnje strukture različitih modela.

3.3 Glavni izazovi prilikom finog podešavanja

Nasuprot brojnim pogodnostima koje fino podešavanje donosi, uvek se treba paziti i na određene probleme koji mogu da se pojave. Neki od njih dati su u nastavku:

- **Katastrofalno zaboravljanje (eng. Catastrophic Forgetting)** – Ovaj problem najčešće se javlja ukoliko koristimo potpuno fino podešavanje u kojem se menja celokupna konfiguracija modela. U ovom slučaju moguće je da se model toliko fokusira na zadate probleme da izgubi moć generalizacije. Rešenje: Elastična konsolidacija težina i progresivno učenje, kao i korišćenje metoda učenja bez ili sa malo primera.

- **Preprilagođavanje na malim skupovima** – Ukoliko je skup podataka premali, model može da zapamti tačan obrazac umesto da se fokusira na efikasnu generalizaciju. Rešenje: Povećanje skupa podataka, različite tehnike regularizacije i rano zaustavljanje.

- **Računska i memorijska zahtevnost** – Kao i u inicijalnom treniranju, i prilikom finog podešavanja ukoliko prosleđujemo modelu veliki skup podataka ili ukoliko se trenira veći deo modela može doći do visoke računske i memorijske zahtevnosti.

  Rešenje: Korišćenje metoda parametarski efikasnog finog podešavanja.

- **Osetljivost na hiperparametre** – Prilikom odabira količine pruženih podataka, kao i količine parametara za treniranje kao i slojeva na kojima će se fino podešavanje ostvariti velika pažnja se mora posvetiti efikasnom odabiru svih ponuđenih parametara.

  Rešenje: Korišćenje Grid Search-a i Bajesovih optimizacija

- **Pristrasnost prema podacima** – Fino podešavanje nad pristrasnim podacima takođe produbljuje pristrasnost.

Rešenje: Korišćenje tehnika za razbijanje pristrasnosti

- **Teškoće prilikom prenošenja znanja** – Neki modeli se teško preprilagođavaju novim problemima ukoliko su njihovi domeni jako različiti.

  Rešenje: Multi-domensko treniranje, korišćenje metoda suprotstavljenog treniranja (eng. adversarial training), nalik na DANN (eng. Domain-Adversarial Neural Networks)

Još jedan od problema je i međusoban uticaj različitih metoda finog podešavanja na model. Ukoliko neke metode pokazuju dobre rezultate kada se primene pojedinačno, to ne znači da će model dobro da se ponaša i kada se one međusobno iskombinuju.

Glava 4

Implementacija rešenja

4.1 Implementacija potpunog finog podešavanja

4.2 Implementacija finog podešavanja klasifikacione glave

4.3 Implementacija finog podešavanja zasnovanog na adapterima

4.4 Implementacija finog podešavanja zasnovanog na matricama niskog ranga

- Dodavanje i prilagodjavanje dodatnog sloja - Nema totalnog zaboravljanja - Potrebne postojece tezine - Zamrzavanje postojecih tezina - postavljanje adaptera - postoji from peft import LoraConfig -

Glava 5

Poređenje rezultata

5.1 Upoređivanje memorijske složenosti

5.2 Upoređivanje vremenske složenosti

5.3 Upoređivanje kompjuterske složenosti

Glava 6

Mogućnosti za dalji napredak

Glava 7

Zaključak

Glava 8

Razrada

Fijuče vetar u šiblju, ledi pasaže i kuće iza njih i gunđa u odžacima. Nidžo, čežnjivo gledaš fotelju, a Đura i Mika hoće poziciju sebi. Ljudi, jazavac Džef trči po šumi glođući neko suho žbunje. Ljubavi, Olga, hajde pođi u Fudži i čut ćeš nježnu muziku srca. Boja vaše haljine, gospođice Džafić, traži da za nju kulučim. Hadži Đera je zaćutao i bacio čežnjiv pogled na šolju s kafom. Džabe se zec po Homolju šunja, čuvar Jožef lako će i tu da ga nađe. Odžačar Filip šalje osmehe tuđoj ženi, a njegova kuća bez dece. Butić Đuro iz Foče ima pun džak ideja o slaganju vaših željica. Džajić odskoči u aut i izbeže đon halfa Pecelja i njegov šamar. Plamte odžaci fabrika a čađave guje se iz njih dižu i šalju noć. Ajšo, lepoto i čežnjo, za ljubav srca moga, dođi u Hadžiće na kafu. Huči šuma, a iza žutog džbuna i panja đak u cveću delje seji frulu. Goci i Jaćimu iz Banje Koviljače, flaša džina i žeđ padahu u istu uru. Džaba što Feđa čupa za kosu Milju, ona juri Živu, ali njega hoće i Daca. Dok je Fehim u džipu žurno ljubio Zagu Čađević, Cile se ušunjao u auto. Fijuče košava nad odžacima a Ilja u gunju žureći uđe u suhu i toplu izbu. Bože, džentlmeni osećaju fizičko gađenje od prljavih šoljica! Dočepaće njega jaka šefica, vođena ljutom srdžbom zlih žena. Pazi, gedžo, brže odnesi šefu taj đavolji ček: njim plaća ceh. Fine džukce ozleđuje bič: odgoj ih pažnjom, strpljivošću. Zamišljao bi kafedžiju vlažnih prstića, crnjeg od čađi. Đače, uštedu plaćaj žaljenjem zbog džinovskih cifara. Džikljaće žalfija između tog busenja i peščanih dvoraca. Zašto gđa Hadžić leči živce: njena ljubav pred fijaskom? Jež hoće peckanjem da vređa ljubičastog džina iz flaše. Džej, ljubičast zec, laže: gađaće odmah pokvašen fenjer. Plašljiv zec hoće jeftinu dinju: grožđe iskamči džabe. Džak je pun žica: ćućeš tad svađu zbog lomljenja harfe. Čuj, džukac Flop bez daha s gađenjem žvaće stršljena. Oh, zadnji šraf na džipu slab: muž gđe Cvijić ljut koči. Šef džabe zvižduće: mlađi

hrt jače kljuca njenog psa. Odbaciće kavgadžija plaštom čađ u željezni fenjer. Deblji krojač: zgužvah smeđ filc u tanjušni džepić. Džo, zgužvaćeš tiho smeđ filc najdeblje krpenjače. Štef, bacih slomljen dečji zvrk u džep gđe Žunjić. Debljoj zgužvah smeđ filc — njen škrt džepčić.

Fijuče vetar u šiblju, ledi pasaže i kuće iza njih i gunđa u odžacima. Nidžo, čežnjivo gledaš fotelju, a Đura i Mika hoće poziciju sebi. Ljudi, jazavac Džef trči po šumi glođući neko suho žbunje. Ljubavi, Olga, hajde pođi u Fudži i čut ćeš nježnu muziku srca. Boja vaše haljine, gospođice Džafić, traži da za nju kulučim. Hadži Đera je zaćutao i bacio čežnjiv pogled na šolju s kafom. Džabe se zec po Homolju šunja, čuvar Jožef lako će i tu da ga nađe. Odžačar Filip šalje osmehe tuđoj ženi, a njegova kuća bez dece. Butić Đuro iz Foče ima pun džak ideja o slaganju vaših željica. Džajić odskoči u aut i izbeže đon halfa Pecelja i njegov šamar. Plamte odžaci fabrika a čađave guje se iz njih dižu i šalju noć. Ajšo, lepoto i čežnjo, za ljubav srca moga, dođi u Hadžiće na kafu. Huči šuma, a iza žutog džbuna i panja đak u cveću delje seji frulu. Goci i Jaćimu iz Banje Koviljače, flaša džina i žeđ padahu u istu uru. Džaba što Feđa čupa za kosu Milju, ona juri Živu, ali njega hoće i Daca. Dok je Fehim u džipu žurno ljubio Zagu Čađević, Cile se ušunjao u auto. Fijuče košava nad odžacima a Ilja u gunju žureći uđe u suhu i toplu izbu. Bože, džentlmeni osećaju fizičko gađenje od prljavih šoljica! Dočepaće njega jaka šefica, vođena ljutom srdžbom zlih žena. Pazi, gedžo, brže odnesi šefu taj đavolji ček: njim plaća ceh. Fine džukce ozleđuje bič: odgoj ih pažnjom, strpljivošću. Zamišljao bi kafedžiju vlažnih prstića, crnjeg od čađi. Đače, uštedu plaćaj žaljenjem zbog džinovskih cifara. Džikljaće žalfija između tog busenja i peščanih dvoraca. Zašto gđa Hadžić leči živce: njena ljubav pred fijaskom? Jež hoće peckanjem da vređa ljubičastog džina iz flaše. Džej, ljubičast zec, laže: gađaće odmah pokvašen fenjer. Plašljiv zec hoće jeftinu dinju: grožđe iskamči džabe. Džak je pun žica: ćućeš tad svađu zbog lomljenja harfe. Čuj, džukac Flop bez daha s gađenjem žvaće stršljena. Oh, zadnji šraf na džipu slab: muž gđe Cvijić ljut koči. Šef džabe zvižduće: mlađi hrt jače kljuca njenog psa. Odbaciće kavgadžija plaštom čađ u željezni fenjer. Deblji krojač: zgužvah smeđ filc u tanjušni džepić. Džo, zgužvaćeš tiho smeđ filc najdeblje krpenjače. Štef, bacih slomljen dečji zvrk u džep gđe Žunjić. Debljoj zgužvah smeđ filc — njen škrt džepčić.

Glava 9

Zaključak

Fijuče vetar u šiblju, ledi pasaže i kuće iza njih i gunđa u odžacima. Nidžo, čežnjivo gledaš fotelju, a Đura i Mika hoće poziciju sebi. Ljudi, jazavac Džef trči po šumi glođući neko suho žbunje. Ljubavi, Olga, hajde pođi u Fudži i čut ćeš nježnu muziku srca. Boja vaše haljine, gospođice Džafić, traži da za nju kulučim. Hadži Đera je zaćutao i bacio čežnjiv pogled na šolju s kafom. Džabe se zec po Homolju šunja, čuvar Jožef lako će i tu da ga nađe. Odžačar Filip šalje osmehe tuđoj ženi, a njegova kuća bez dece. Butić Đuro iz Foče ima pun džak ideja o slaganju vaših željica. Džajić odskoči u aut i izbeže đon halfa Pecelja i njegov šamar. Plamte odžaci fabrika a čađave guje se iz njih dižu i šalju noć. Ajšo, lepoto i čežnjo, za ljubav srca moga, dođi u Hadžiće na kafu. Huči šuma, a iza žutog džbuna i panja đak u cveću delje seji frulu. Goci i Jaćimu iz Banje Koviljače, flaša džina i žeđ padahu u istu uru. Džaba što Feđa čupa za kosu Milju, ona juri Živu, ali njega hoće i Daca. Dok je Fehim u džipu žurno ljubio Zagu Čađević, Cile se ušunjao u auto. Fijuče košava nad odžacima a Ilja u gunju žureći uđe u suhu i toplu izbu. Bože, džentlmeni osećaju fizičko gađenje od prljavih šoljica! Dočepaće njega jaka šefica, vođena ljutom srdžbom zlih žena. Pazi, gedžo, brže odnesi šefu taj đavolji ček: njim plaća ceh. Fine džukce ozleđuje bič: odgoj ih pažnjom, strpljivošću. Zamišljao bi kafedžiju vlažnih prstića, crnjeg od čađi. Đače, uštedu plaćaj žaljenjem zbog džinovskih cifara. Džikljaće žalfija između tog busenja i peščanih dvoraca. Zašto gđa Hadžić leči živce: njena ljubav pred fijaskom? Jež hoće peckanjem da vređa ljubičastog džina iz flaše. Džej, ljubičast zec, laže: gađaće odmah pokvašen fenjer. Plašljiv zec hoće jeftinu dinju: grožđe iskamči džabe. Džak je pun žica: ćućeš tad svađu zbog lomljenja harfe. Čuj, džukac Flop bez daha s gađenjem žvaće stršljena. Oh, zadnji šraf na džipu slab: muž gđe Cvijić ljut koči. Šef džabe zvižduće: mlađi

hrt jače kljuca njenog psa. Odbaciće kavgadžija plaštom čađ u željezni fenjer. Deblji krojač: zgužvah smeđ filc u tanjušni džepić. Džo, zgužvaćeš tiho smeđ filc najdeblje krpenjače. Štef, bacih slomljen dečji zvrk u džep gđe Žunjić. Debljoj zgužvah smeđ filc — njen škrt džepčić.

Fijuče vetar u šiblju, ledi pasaže i kuće iza njih i gunđa u odžacima. Nidžo, čežnjivo gledaš fotelju, a Đura i Mika hoće poziciju sebi. Ljudi, jazavac Džef trči po šumi glođući neko suho žbunje. Ljubavi, Olga, hajde pođi u Fudži i čut ćeš nježnu muziku srca. Boja vaše haljine, gospođice Džafić, traži da za nju kulučim. Hadži Đera je zaćutao i bacio čežnjiv pogled na šolju s kafom. Džabe se zec po Homolju šunja, čuvar Jožef lako će i tu da ga nađe. Odžačar Filip šalje osmehe tuđoj ženi, a njegova kuća bez dece. Butić Đuro iz Foče ima pun džak ideja o slaganju vaših željica. Džajić odskoči u aut i izbeže đon halfa Pecelja i njegov šamar. Plamte odžaci fabrika a čađave guje se iz njih dižu i šalju noć. Ajšo, lepoto i čežnjo, za ljubav srca moga, dođi u Hadžiće na kafu. Huči šuma, a iza žutog džbuna i panja đak u cveću delje seji frulu. Goci i Jaćimu iz Banje Koviljače, flaša džina i žeđ padahu u istu uru. Džaba što Feđa čupa za kosu Milju, ona juri Živu, ali njega hoće i Daca. Dok je Fehim u džipu žurno ljubio Zagu Čađević, Cile se ušunjao u auto. Fijuče košava nad odžacima a Ilja u gunju žureći uđe u suhu i toplu izbu. Bože, džentlmeni osećaju fizičko gađenje od prljavih šoljica! Dočepaće njega jaka šefica, vođena ljutom srdžbom zlih žena. Pazi, gedžo, brže odnesi šefu taj đavolji ček: njim plaća ceh. Fine džukce ozleđuje bič: odgoj ih pažnjom, strpljivošću. Zamišljao bi kafedžiju vlažnih prstića, crnjeg od čađi. Đače, uštedu plaćaj žaljenjem zbog džinovskih cifara. Džikljaće žalfija između tog busenja i peščanih dvoraca. Zašto gđa Hadžić leči živce: njena ljubav pred fijaskom? Jež hoće peckanjem da vređa ljubičastog džina iz flaše. Džej, ljubičast zec, laže: gađaće odmah pokvašen fenjer. Plašljiv zec hoće jeftinu dinju: grožđe iskamči džabe. Džak je pun žica: ćućeš tad svađu zbog lomljenja harfe. Čuj, džukac Flop bez daha s gađenjem žvaće stršljena. Oh, zadnji šraf na džipu slab: muž gđe Cvijić ljut koči. Šef džabe zvižduće: mlađi hrt jače kljuca njenog psa. Odbaciće kavgadžija plaštom čađ u željezni fenjer. Deblji krojač: zgužvah smeđ filc u tanjušni džepić. Džo, zgužvaćeš tiho smeđ filc najdeblje krpenjače. Štef, bacih slomljen dečji zvrk u džep gđe Žunjić. Debljoj zgužvah smeđ filc — njen škrt džepčić.

Bibliografija

[1] OpenAI. GPT-4 Technical Report, 2023. https://arxiv.org/abs/2303.08774.

[2] Meta AI. Introducing llama: A foundational, 65-billion-parameter language model, 2023. https://ai.meta.com/blog/large-language-model-llama-meta-ai/.

[3] Abi Aryan, Aakash Kumar Nain, Andy McMahon, Lucas Augusto Meyer, Harpeet Singh Sahota. The Costly Dilemma: Are Large Language Models the Pay-Day Loans of MAchine Leaning? 2023. https://abiaryan.com/assets/EMNLP%20Submission_Non-Anon.pdf/.

[4] Hugging Face. Hugging Face: The AI Community Building the Future, 2023. https://huggingface.co/.

[5] Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[6] Raffel, Colin and Shinn, Adam and Cohn, Trevor and others. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[7] Wenxuan Wang, Jifan Yu, and Xiang Ren. Parameter-efficient tuning makes a good classification head. *arXiv preprint arXiv:2210.16771*, 2022.

[8] Parameter-Efficient Transfer Learning for NLP, author=Houlsby, Neil and Giurgiu, Andrei and Jastrzebski, Stanislaw and Morrone, Brianna and de Laroussilhe, Quentin and Gesmundo, Andrea and Attariyan, Mona and Gelly, Sylvain, booktitle=Proceedings of the 36th International

Conference on Machine Learning (ICML), pages=2790–2799, year=2019, url=https://arxiv.org/abs/1902.00751.

[9] Edward J. Hu and Yelong Shen and Phillip Wallis and Zeyuan Allen-Zhu and Yuanzhi Li and Shean Wang and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[10] He, Ruibin and Anastasopoulos, Antonios and Neubig, Graham. A Survey on Parameter Efficient Transfer Learning for NLP. *arXiv preprint arXiv:2203.06904*, 2022.

[11] Bing Liu. *Sentiment Analysis and Opinion Mining*. 2012.

[12] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[13] Prashant Johri, Sunil Kumar Khatri, Ahmad Al-Taani, Munish Sabharwal, Shakhzod Suvanov, and Avneesh Chauhan. *Natural Language Processing: History, Evolution, Application, and Future Work*, pages 365–375. 01 2021.

[14] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198. Association for Computational Linguistics, July 2020.

[15] Paul Semaan. Natural language generation: an overview. *J Comput Sci Res*, 1(3):50–57, 2012.

[16] Jin Guo. Critical tokenization and its properties. *Computational Linguistics*, 23(4):569–596, 1997.

[17] Hugging Face. Tokenizers — hugging face nlp course. `https://huggingface.co/learn/nlp-course/en/chapter2/4`, 2023. Accessed: 2025-01-28.

[18] Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Schulze Buschhoff, et al. Tokenizer choice for llm training: Negligible or crucial? *arXiv preprint arXiv:2310.08754*, 2023.

[19] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N. and Kaiser, Łukasz and Polosukhin, Illia. Attention is All You Need. *Advances in Neural Information Processing Systems (NeurIPS 2017)*, 2017.

[20] Daniel Svozil, Vladimír Kvasnicka, and Jiri Pospichal. Introduction to multilayer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1):43–62, 1997.

[21] Ruibin Xiong, Yi Yang, Di He, Kai Zheng, Shuo Zheng, Chao Xing, Hang Zhang, Yelong Lan, Liwei Wang, Tie-Yan Liu, and Maosong Sun. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, 2020.

[22] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*, 2019.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *arXiv preprint arXiv:1503.02531*, 2015.

[25] Ronen Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, April 2013.

[26] Benaissa Azzeddine Rachid, Harbaoui Azza, and Ben Ghezala Henda. Sentiment analysis approaches based on granularity levels. In *Proceedings of the 14th International Conference on Web Information Systems and Technologies*, volume 1, pages 324–331, 2018.

[27] Doaa Mohey El-Din Mohamed Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018.

[28] Prabha PM Surya and B Subbulakshmi. Sentimental analysis using naive bayes classifier. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pages 1–5, 2019.

[29] Munir Ahmad, Shabib Aftab, and Iftikhar Ali. Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, 177(5):25–29, 2017.

[30] Xi Ouyang, Pan Zhou, Cheng Hua Li, and Lijun Liu. Sentiment analysis using convolutional neural network. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2359–2364, 2015.

[31] Lilis Kurniasari and Arif Setyanto. Sentiment analysis using recurrent neural network. *Journal of Physics: Conference Series*, 1471(1):012018, feb 2020.

[32] GSN Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, Mounika Bagadi, and Mounika Belusonti. Text based sentiment analysis using lstm. *Int. J. Eng. Res. Tech. Res*, 9(05):299–303, 2020.

[33] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[34] Tanjim Ul Haque, Nudrat Nawal Saber, and Faisal Muhammad Shah. Sentiment analysis on large scale amazon product reviews. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6, 2018.

[35] Himanshu Pal and Bharat Bhushan. Sentiment analysis on twitter dataset using voting classifier. In *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, volume 1, pages 1–6, 2024.

[36] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[37] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[38] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[39] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.

Biografija autora

Vuk Stefanović Karadžić (*Tršić, 26. oktobar/6. novembar 1787. — Beč, 7. februar 1864.*) bio je srpski filolog, reformator srpskog jezika, sakupljač narodnih umotvorina i pisac prvog rečnika srpskog jezika. Vuk je najznačajnija ličnost srpske književnosti prve polovine XIX veka. Stekao je i nekoliko počasnih mastera. Učestvovao je u Prvom srpskom ustanku kao pisar i činovnik u Negotinskoj krajini, a nakon sloma ustanka preselio se u Beč, 1813. godine. Tu je upoznao Jerneja Kopitara, cenzora slovenskih knjiga, na čiji je podsticaj krenuo u prikupljanje srpskih narodnih pesama, reformu ćirilice i borbu za uvođenje narodnog jezika u srpsku književnost. Vukovim reformama u srpski jezik je uveden fonetski pravopis, a srpski jezik je potisnuo slavenosrpski jezik koji je u to vreme bio jezik obrazovanih ljudi. Tako se kao najvažnije godine Vukove reforme ističu 1818., 1836., 1839., 1847. i 1852.