Prepoznavanje teksta generiranog velikim jezičnim modelom

Fain, Maria

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:792042

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-22**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 514

PREPOZNAVANJE TEKSTA GENERIRANOG VELIKIM JEZIČNIM MODELOM

Maria Fain

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 514

PREPOZNAVANJE TEKSTA GENERIRANOG VELIKIM JEZIČNIM MODELOM

Maria Fain

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 514

Pristupnica:	**Maria Fain (0036522285)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	izv. prof. dr. sc. Goran Delač

Zadatak: **Prepoznavanje teksta generiranog velikim jezičnim modelom**

Opis zadatka:

Proučiti i opisati postupke klasifikacije teksta zasnovane na modelima dubokih neuronskih mreža. Pritom, posebno uzeti u obzir modele zasnovane na transformerima, poput modela BERT. Prikupiti skup podataka koji sadrži izvorne tekstove o određenoj skupini tema popraćene tekstovima koji su umjetno stvoreni primjenom velikog jezičnog modela. Skup podataka prikladno pripremiti te provesti podjelu na podskupove za učenje i testiranje. Provesti primjereni postupak vrednovanja klasifikatora.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod **3**

2. Veliki jezični modeli **5**

2.1. Radovi povezani s tematikom prepoznavanja teksta generiranog velikim jezičnim modelom 6

2.2. Problematika klasifikacije teksta generiranog velikim jezičnim modelima 7

3. Programski razvoj klasifikatora namijenjenih za prepoznavanje teksta generiranog velikim jezičnim modelom **9**

3.1. Opis korištenog skupa podataka 9

3.1.1. Dodatni podatci za ispitivanje 11

3.1.2. Pretprocesiranje podataka 12

3.1.3. Podjela podataka 14

3.2. Opis korištenih knjižnica 14

4. Opis programski ostvarenih klasifikatora namijenjenih za prepoznavanje teksta generiranog velikim jezičnim modelom **16**

4.1. Modeli zasnovani na transformerima 16

4.2. Klasifikator BERT 18

4.2.1. Arhitektura programski ostvarenog klasifikatora BERT 20

4.3. Klasifikator RoBERTa 22

4.3.1. Arhitektura programski ostvarenog klasifikatora RoBERTa 23

4.4. Klasifikator DistilBERT 25

4.4.1. Arhitektura programski ostvarenog klasifikatora DistilBERT . . . 27

5. Opis metrika za vrednovanje uspješnosti prepoznavanja teksta generira-

nog velikim jezičnim modelom **29**
5.1. Matrica zabune 29
5.2. Točnost 30
5.3. Preciznost 30
5.4. Odziv 31
5.5. Mjera F1 31
5.6. ROC krivulja i površina ispod ROC krivulje 32

6. Rezultati i rasprava **34**
6.1. Rezultati klasifikatora BERT 34
6.1.1. Rezultati klasifikatora BERT nad dodatnim podatcima za ispitivanje 36
6.2. Rezultati klasifikatora RoBERTa 38
6.2.1. Rezultati klasifikatora RoBERTa nad dodatnim podatcima za ispitivanje 40
6.3. Rezultati klasifikatora DistilBERT 41
6.3.1. Rezultati klasifikatora DistilBERT nad dodatnim podatcima za ispitivanje 43
6.4. Rasprava 45

7. Zaključak **48**

Literatura **49**

Sažetak **53**

Abstract **54**

1. Uvod

Jedno od područja umjetne inteligencije je područje obrade prirodnog jezika koje se bavi zadatcima za koje su potrebni podatci u obliku teksta. U sklopu ovog područja razvijeni su brojni jezični modeli tijekom povijesti. Novim otkrićima i napretkom u području obrade prirodnog jezika, razvijeni su i veliki jezični modeli koji imaju veći broj parametara i bolje sposobnosti rješavanja zadataka s tekstualnim podatcima. Njihovi zadatci uključuju probleme poput odgovaranja na pitanja, prevođenja teksta, klasifikacije teksta, generiranja teksta i slično. Razvoj ovako naprednih modela koji su javno dostupni učinio je njihovo korištenje sve učestalijim. Ljudi su tako počeli i zlouporabljivati jezične modele za generiranje teksta kojeg je bilo potrebno samostalno napisati. S porastom razina točnosti velikih jezičnih modela, generirani tekst postalo je sve teže za uočiti ljudskom intervencijom što je počelo stvarati probleme u raznim domenama, pogotovo u akademskom svijetu. Zbog toga se u posljednjih par godina intenzivno radi na programskim ostvarenjima sustava koji će imati mogućnost prepoznavanja teksta koji je generiran nekim velikim jezičnim modelom.

Cilj ovog rada je ostvariti i vrednovati sustav takav da će imati mogućnost klasificiranja teksta na tekst generiran velikim jezičnim modelom i tekst koji je napisao čovjek. Programski su ostvarena tri klasifikatora čiji se rad detaljno vrednovao i potom usporedio. Za programsko ostvarenje koristili su se modeli BERT, RoBERTa i DistilBERT zasnovani na transformerima koji u svojoj arhitekturi koriste samo enkodere (engl. *encoder-only models*). Arhitektura ovih modela zasnovana je na metodama dubokog učenja i mehanizmima pažnje (engl. *attention mechanisms*). Skup podataka *„GPT Wiki Intro”* koji se koristio za učenje i ispitivanje modela sadrži uvode raznih tema s internetske stranice *„Wikipedia”* i uvode koji su generirani velikim jezičnim modelom GPT *Curie*. Nakon što su modeli naučeni s podatcima iz skupa podataka *„GPT Wiki Intro”*, vrednovani su raz-

nim metrikama. Najvišu razinu točnosti od čak 0.965 imao je klasifikator RoBERTa, dok je klasifikator DistilBERT imao točnost od 0.963, a klasifikator BERT 0.956. Klasifikatori su dodatno vrednovani nad skupovima podataka koji sadrže tekstove generirane velikim jezičnim modelima PaLM, Falcon i LLaMA. Najbolje rezultate nad ovim podatcima ponovno je pokazao klasifikator RoBERTa s mjerom odziva koja iznosi 0.9.

Rad je organiziran na sljedeći način: u poglavlju 2. opisani su općenito veliki jezični modeli i problematika klasifikacije teksta. Skup podataka koji se koristio pri modeliranju, dodatni skupovi za ispitivanje te korištene knjižnice opisani su u poglavlju 3. Opis programski ostvarenih klasifikatora i njihovih arhitektura dan je u poglavlju 4. Zatim, metrike korištene za vrednovanje klasifikatora opisane su u poglavlju 5., a rezultati i diskusija navedeni su u poglavlju 6. Konačno, zaključak rada naveden je u poglavlju 7.

2. Veliki jezični modeli

Obrada prirodnog jezika (engl. *natural language processing - NLP*) je jedno od glavnih područja primjene umjetne inteligencije. Cilj ove discipline je učiti strojeve da razumiju, sažimaju i generiraju prirodni jezik. Neki od problema i primjena koje područje obrade prirodnog jezika obuhvaća su klasifikacija teksta, analiza sentimenta teksta, prevođenje strojnog jezika, odgovaranje na pitanja, generiranje teksta, itd. U cilju da se ovi problemi riješe, brojni jezični modeli su razvijeni kroz povijest. Evolucijom jezičnih modela i područja obrade prirodnog jezika, nastali su veliki jezični modeli (engl. *large language models - LLMs*). Oni se razlikuju od manjih jezičnih modela u broju parametara i mogućnostima procesiranja i generiranja velikih količina teksta.

Veliki jezični modeli obuhvaćaju najnovije tehnologije u području obrade prirodnog jezika, a i općenito područja umjetne inteligencije. To su modeli koji koriste tehnologije dubokog učenja i velike skupove tekstualnih podataka da bi mogli razumjeti, sažeti, predvidjeti i/ili generirati novi sadržaj s visokom razinom preciznosti. Mnogi moderni veliki jezični modeli zasnovani su na transformerima. Transformeri su arhitekture modela koje se u potpunosti oslanjaju na mehanizme pažnje (engl. *attention mechanisms*) koji povlače globalne ovisnosti između ulaznih i izlaznih podataka, a uvedeni su 2017. s [1]. Od tada su u kratkom vremenskom roku razvijeni neki od trenutno najpopularnijih i preciznijih velikih jezičnih modela, poput BERT-a koji je predstavljen u [2]. Također, trenutno vrlo popularni modeli su generativni modeli, točnije GPT-ovi (engl.*Generative Pre-trained Transformers*). To su veliki jezični modeli koji su prethodno trenirani nad velikom količinom tekstualnih podataka dostupnih na internetu koji im omogućavaju da nauče kako je strukturiran prirodan jezik. Mogu se koristiti za generiranje, prevođenje ili sažimanje teksta. Nakon što je tvrtka *„OpenAI"* objavila svoj besplatan i javno dostupan alat *ChatGPT*, popularnost i korištenje velikih jezičnih modela naglo je poraslo ne

samo u tehničkom svijetu, već i u cijeloj javnosti. *ChatGPT* je model umjetne inteligencije zasnovan na arhitekturi GPT-3.5 koji je sposoban odgovarati na pitanja i generirati tekstualni sadržaj o temama iz raznih područja. Zbog lake uporabe i široke dostupnosti, alati koji u pozadini koriste velike jezične modele, poput *ChatGPT*-a, počeli su se koristiti za pomoć pri svakodnevnim ljudskim zadatcima.

Kao i sa svakim novim javno dostupnim alatom, postoji mogućnost njegove uporabe u krive svrhe. Tako se javio problem korištenja velikih jezičnih modela za generiranje plagiranog ili lažnog sadržaja. Najveći problemi javljaju se u obrazovnom i akademskom svijetu gdje učenici i studenti koriste generativne modele za pisanje svojih radova ili eseja. S obzirom na to da je generirani tekst sve teže otkriti ljudskom intervencijom, mnoge obrazovne ustanove trenutno nemaju rješenja za prepoznavanje radova učenika koji varaju sustav. Također, u [3] naglašava se i problem lažnih vijesti u medijima koje se generiraju koristeći velike jezične modele.

2.1. Radovi povezani s tematikom prepoznavanja teksta generiranog velikim jezičnim modelom

S obzirom na to da je pristup velikim generativnim jezičnim modelima omogućen tek nedavno, problemi koji nastaju njihovim korištenjem tek se istražuju. Postoji samo nekolicina objavljenih radova kojima je cilj detektirati sadržaj generiran velikim jezičnim modelom. Ipak, u prethodnoj i tekućoj godini, ljudi su postali svjesni problema koji su se stvorili u sklopu raznih domena te su počeli razvijati modele kojima je cilj detektirati tekstove koje je napisao veliki jezični model i diferencirati ih od onih tekstova koje su zapravo napisali ljudi. Tako su Koike, Kaneko i Okazaki u [4] razvili model koji detektira eseje generirane velikim jezičnim modelima. U [5] Munyer i ostali tvrde da klasifikatori imaju prevelika ograničenja pa su razvili metodologiju zasnovanu na dubokom učenju koja može utvrditi izvor teksta koristeći njegov vodeni žig (engl.*watermark*). Također, tvrtka *„OpenAI"* je objavila svoj novi alat *„AI Content Detector"* za prepoznavanje sadržaja kojeg je generirala umjetna inteligencija. Na taj način žele pomoći u suzbijanju korištenja generiranih tekstova u krive svrhe. Ovom alatu prethodio je njihov *„AI Classifier"*, klasifikator za prepoznavanje teksta generiranog umjetnom inteligencijom. Nakon što je utvrđeno da taj klasifikator ipak ima veliku razinu nesigurnosti, alat je povučen s

njihovih službenih stranica.

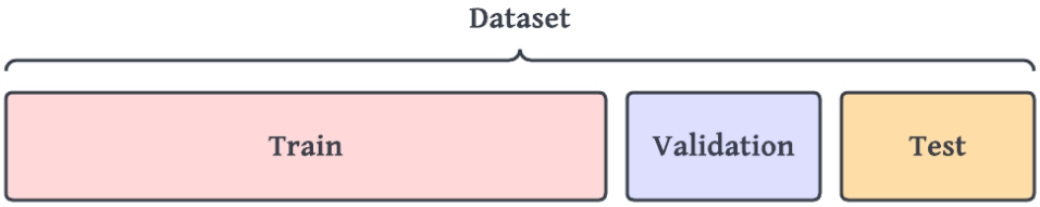
2.2. Problematika klasifikacije teksta generiranog velikim jezičnim modelima

Prije opisivanja programskog rješenja i podataka korištenih za ostvarenje rješenja, potrebno je opisati i definirati neke pojmove koji će se koristiti kroz ostatak rada, a koji su sastavni dio područja obrade prirodnog jezika i strojnog učenja općenito.

Klasifikacija teksta dio je problematike kojom se bavi područje obrade prirodnog jezika. Radi se o nadziranom strojnom učenju, gdje svaki primjer ima svoju oznaku, odnosno klasu. Tekstu se pridodaju oznake (klase) te se predviđa koji tekst ima koju oznaku (klasu). U slučaju detekcije teksta generiranog velikim jezičnim modelom, postoje dvije oznake (generirani i ne generirani tekst) što znači da se radi o binarnoj klasifikaciji. Model strojnog učenja koji se koristi za klasifikaciju naziva se klasifikator.

Kao i kod svakog problema strojnog učenja, modelima su potrebni podatci da bi mogli donositi zaključke. Skup podataka koji će se koristiti za modeliranje potrebno je podijeliti na tri skupa: skup za učenje (engl. *training set*), skup za validaciju (engl. *validation set*) te skup za ispitivanje ili ocijenjivanje (engl. *testing set*). Ovi skupovi su međusobno disjunktni, odnosno nemaju zajedničkih primjera. Skup za učenje koristi se da bi se model naučio donositi zaključke na podatcima. Proces učenja modela odvija se u više faza koje se nazivaju epohe, a njihov broj se unaprijed određuje. Skup za validaciju koristi se da bi se ispitao rad modela nakon svake epohe što pomaže u donošenju zaključaka oko promjene parametara modela. Nakon što je model naučen, njegov rad se vrednuje koristeći skup podataka za ispitivanje i različite metrike koje su detaljno opisane u poglavlju 5. Bitno je da se rad modela ispituje nad podatcima koje model dotada nije susreo. Uobičajena podjela podataka na skupove za učenje, validaciju i ispitivanje vrši se u omjeru 70:15:15, ali ovisi o količini podataka. Primjer podjele skupa podataka na tri manja skupa prikazana je slikom 2.1.

Pri procesu učenja, svaki model ima unaprijed definiranu empirijsku pogrešku i funkciju gubitka. Empirijska pogreška ukazuje na to koliko model dobro radi klasifikacije na određenom dijelu skupa podataka. To je pogreška kojom se vrednuje skup podataka za



Slika 2.1. Primjer podjele skupa podataka na skup za učenje, validaciju i ispitivanje [6]

validaciju nakon svake epohe. Funkcija gubitka izračunava iznos pogreške koju model ima za svaki pojedinačni primjer. Ona ukazuje na to za koliko je model uvećao ukupnu pogrešku za taj pojedini primjer.

U području obrade prirodnog jezika postoje stručni izrazi koji se najčešće pojavljuju pri obradi podataka za modele. Tekstualni podatci se moraju prikladno obraditi prije modeliranja u fazi zvanoj pretprocesiranje (engl. *preprocessing*). Neki od pojmova koji se koriste samo u toj fazi objašnjeni su u poglavlju 3.1.2. Ostali često korišteni pojmovi definirat će se u ovom poglavlju. Korpus (engl. *corpus*) je skup tekstova koje obuhvaća skup podataka koji se koristi za modeliranje. U području obrade prirodnog jezika, tokeni predstavljaju jedinice od kojih se sastoje rečenice. Tokeni mogu biti numerički znakovi, interpunkcijski znakovi ili riječi.

3. Programski razvoj klasifikatora namijenjenih za prepoznavanje teksta generiranog velikim jezičnim modelom

Cilj ovog rada je ostvariti programsko rješenje u obliku klasifikatora koji će imati mogućnost prepoznavanja teksta generiranog velikim jezičnim modelom. Za klasifikaciju će se koristiti više velikih jezičnih modela zasnovanih na dubokom učenju koji će se detaljno opisati. Modeli koji će se koristiti za klasifikaciju su BERT, RoBERTa te DistilBERT. Nakon treniranja ovih modela s prikladnim podatcima, njihovi rezultati će se usporediti i diskutirati. Cilj svakog klasifikatora je da za ulazni tekst napravi predikciju o tome je li on generiran velikim jezičnim modelom ili nije.

3.1. Opis korištenog skupa podataka

Da bi modeli umjetne inteligencije radili što bolje predikcije, potrebna im je što veća količina prikladnih podataka nad kojima bi se provelo učenje. Svim modelima koji su korišteni za programsko ostvarenje prepoznavanja teksta generiranog velikim jezičnim modelom potrebni su tekstualni ulazni podaci koji se trebaju pravilno obraditi prije treniranja modela. Skup podataka koji je korišten za treniranje modela naziva se *„GPT Wiki Intro”* i preuzet je sa službene *„Hugging Face”* internetske stranice namijenjene razvoju strojnog učenja i modela zasnovanih na transformerima [7].

„GPT Wiki Intro” je skup podataka koji sadrži tekstove uvoda raznih tema sa stranice *„Wikipedia”* i tekstove na istu temu koji su generirani GPT modelom *Curie*. Svi tekstovi su na engleskom jeziku. Najčešće riječi u skupu podataka prikazane su slikom 3.1. na način da veličina riječi odgovara njezinoj učestalosti (frekvenciji) u skupu podataka.

Slika 3.1. Prikaz najčešćih riječi u skupu podataka

„GPT Wiki Intro” sadrži i dodatne značajke poput naslova teksta, duljine originalnog teksta, duljine generiranog teksta, URL-a stranice s koje je preuzet originalni tekst, itd. Skup podataka veličine je 150 tisuća redaka, odnosno postoji 300 tisuća tekstova jer svaki redak u skupu podataka sadrži uvodni tekst sa stranice *„Wikipedia”* i generirani tekst na istu temu. Primjer prvih dva retka iz skupa podataka *„GPT Wiki Intro”* prikazan je slikom 3.2.

	id	url	title	wiki_intro	generated_intro	title_len	wiki_intro_len	generated_intro_len	prompt	generated_text	prompt_tokens
0	63064638	https://en.wikipedia.org/wiki/Sexhow%20railway...	Sexhow railway station	Sexhow railway station was a railway station b...	Sexhow railway station was a railway station l...	3	174	78	200 word wikipedia style introduction on 'Sexh...	located in the town of Sexhow, on the Cumbria...	25
1	279621	https://en.wikipedia.org/wiki/Eti%C3%A4inen	Etiäinen	In Finnish folklore, all places and things, an...	In Finnish folklore, all places and things, an...	1	187	80	200 word wikipedia style introduction on 'Etiä...	animate or inanimate, have a spirit or "etiäi...	26

Slika 3.2. Prva dva retka *„GPT Wiki Intro”* skupa podataka

Da bi se ovaj skup podataka mogao koristiti za klasifikaciju teksta, potrebno ga je preoblikovati na način da svaki redak sadrži tekst i njegovu pripadajuću oznaku. Ostale značajke nisu bitne za modeliranje. Moguće oznake teksta su „generiran” (engl.*”generated”*), označavajući tekst koji je generiran velikim jezičnim modelom, te „nije generiran” (engl.*”not generated”*), označavajući tekst koji nije generiran velikim jezičnim modelom, odnosno tekst koji je napisao čovjek. Ovakvim preoblikovanjem originalnog *„GPT Wiki Intro”* skupa podataka dobivamo novi skup podataka veličine 300 tisuća redaka (jer je originalni skup podataka u svakom retku sadržavao i generirani i originalni tekst). Primjer

prvih nekoliko redaka iz novog, preoblikovanog skupa podataka prikazan je slikom 3.3. Bitno je napomenuti da se tekstovi i njihove oznake moraju dodatno preoblikovati da bi bili prikladni za treniranje modela. Nad tekstovima se obavlja tzv. pretprocesiranje, a oznake se transformiraju u numerički oblik. Više o tome bit će objašnjeno u potpoglavlju 3.1.2. Dodatno pripremanje tekstova za treniranje modela razlikuje se za svaki model pa će biti detaljnije objašnjeno u poglavljima vezanim uz same modele, 4.

	id	title	text	label
0	63064638	Sexhow railway station	Sexhow railway station was a railway station b...	not generated
1	18704707	Sexhow railway station	Sexhow railway station was a railway station l...	generated
2	279621	Etiäinen	In Finnish folklore, all places and things, an...	not generated
3	58254742	Etiäinen	In Finnish folklore, all places and things, an...	generated

Slika 3.3. Prva četiri retka transformiranog skupa podataka

Zbog računalne i vremenske složenosti zadataka treniranja modela, skup podataka je prepolovljen na način da sadrži jednak broj generiranih i negeneriranih tekstova. U cilju da se smanji pristranost, uzeti su tekstovi s različitim naslovima, odnosno s različitim temama. Finalni skup podataka koji je korišten za učenje, validaciju i ispitivanje sadrži 150 tisuća redaka, odnosno tekstova.

Ukratko, klasifikatori na ulaz primaju tekst transformiran u oblik koji je prikladan za pojedini model. Svaki klasifikator kao izlaz vraća informaciju o tome je li ulazni tekst bio generiran ili nije.

3.1.1. Dodatni podatci za ispitivanje

Uz skup podataka za ispitivanje koji je podskup podataka *„GPT Wiki Intro"* korištenih za modeliranje klasifikatora, provedeno je dodatno ispitivanje rada modela nad podatcima koji su prikupljeni iz drugih izvora. Iako se odvajanjem dijela originalnih podataka za skup za ispitivanje postiže vrednovanje modela nad dotad neviđenim podatcima, postoji mogućnost da je skup koji se koristio za ispitivanje u nekom postotku sličan skupu koji se koristio za učenje modela. U cilju da se utvrdi kako klasifikatori rade nad potpuno drugačijim podatcima, prikupljeni su podatci koji uključuju tekstove iz obiju klasa. S obzirom na to da su modeli klasifikatora trenirani nad podatcima koje je generirao samo jedan tip velikih jezičnih modela, a to je model Curie, prikupljeni su dodatni tekstovi

koje su generirali različiti veliki jezični modeli. S vrednovanjem klasifikatora nad ovim skupom podataka moguće je procijeniti jesu li oni prenaučeni, odnosno hoće li raditi s visokom razinom preciznosti i nad tekstovima koje su generirali i drugi veliki jezični modeli.

Skup podataka koji služi za dodatnu provjeru klase negeneriranih tekstova prikupljen je sa *Kaggle* natjecanja na temu *LLM - Detect AI Generated Text*. Sastoji se od 1375 eseja koje su napisali studenti te 3 eseja generiranih nekim velikim jezičnim modelom.

Skup podataka koji služi za dodatnu provjeru klase generiranih tekstova sadrži tekstove koje su generirali veliki jezični modeli PaLM (engl. *Pathways Language Model*) kojeg je razvio *Google* [8], Falcon [9] i LLaMA (engl. *Large Language Model Meta AI*) kojeg je razvila *Meta AI* [10]. Ovi modeli obuhvaćaju neke od najnovijih i najpreciznijih tehnologija u području obrade prirodnog jezika. Skup podataka sadrži 1384 tekstova generiranih modelom PaLM, 1055 tekstova generiranih modelom Falcon te 1172 tekstova generiranih modelom LLaMA.

3.1.2. Pretprocesiranje podataka

Jedan od neizostavnih koraka pri rješavanju svakog problema iz područja obrade prirodnog jezika je pretprocesiranje ili „čišćenje" ulaznih podataka, odnosno teksta. Ovaj korak obavlja se prije samog procesa modeliranja.

Pri procesu „čišćenja" podataka sva velika slova pretvorena su u mala, uklonjene su poveznice, svi interpunkcijski znakovi i veće praznine u tekstu. Tekstovi su tokenizirani i uklonjene su tzv. stop riječi (engl. *stopwords*). Proces tokenizacije označava podjelu teksta na tokene, odnosno dijelove rečenica. Stop riječi su riječi koje se učestalo pojavljuju u određenom jeziku, a ne pridodaju na važnosti i značenju samog teksta. Iz tog razloga je uobičajeno da se one u potpunosti uklone iz teksta, olakšavajući proces obrade teksta i samo modeliranje. Neke od najčešćih stop riječi u engleskom jeziku su: *the, a, is, an, are*... Nakon uklanjanja tokena stop riječi, obavljen je proces lematizacije (engl. *lemmatization*). Lematizacija je postupak svođenja riječi na njihov osnovni oblik, odnosno lemu. Uz lematizaciju, postoji i postupak zvan korjenovanje (engl. *stemming*). Pri tom postupku riječi se svode na njihov korijenski oblik tako što se uklone svi prefiksi ili sufiksi riječi. Proces korjenovanja je brži od lematizacije, ali pruža manju preciznost jer ne

uzima u obzir kontekst te pri uklanjanju nastavaka može doći do gubitka u značenju riječi. Primjer razlike postupka lematizacije i korjenovanja prikazan je slikom 3.4. Nakon lematizacije izbačeni su tokeni koji su imali manje od dva znaka ili slova.

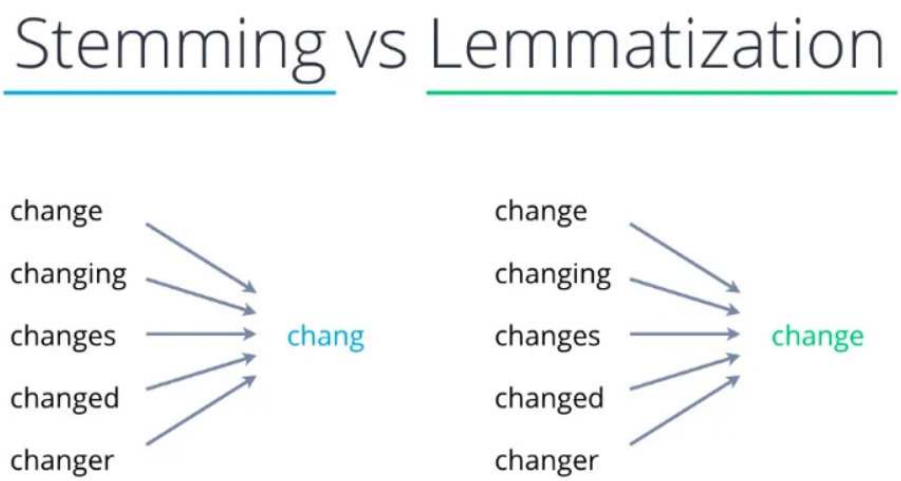
Oznake tekstova enkodirane su na način da je generirani tekst predstavljen brojem 1, a negenerirani tekst brojem 0. Zbog ovih oznaka klasa enkodirana s brojem 1 često se naziva pozitivna klasa, a klasa enkodirana s brojem 0 negativna klasa.

U nastavku slijedi primjer rečenice iz korištenog skupa podataka prije pretprocesiranja i nakon. Rečenica prije pretprocesiranja:

```
In Finnish folklore, all places and things, and also human beings, have
a haltija (a genius, guardian spirit) of their own .
```

Rečenica nakon pretprocesiranja:

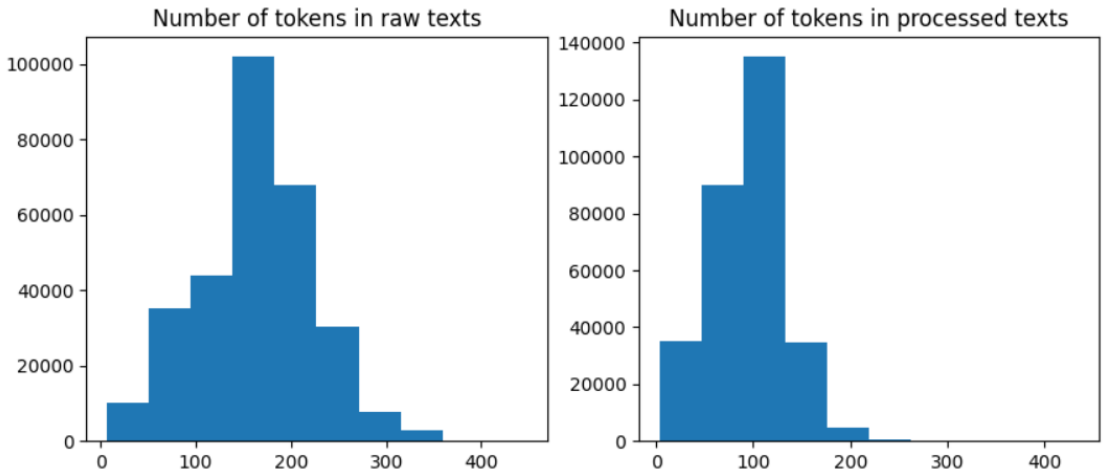
```
finnish folklore place thing also human being haltija genius guardian
spirit
```



Slika 3.4. Postupci korjenovanja (engl.*stemming*) i lematizacije [11]

Pri eksplorativnoj analizi podataka (engl. *Exploratory Data Analysis - EDA*), analizirane su neke od značajki tekstova u korpusu prije i nakon faze pretprocesiranja. Pretprocesiranjem tekstova mijenjaju se broj tokena u tekstu, duljina znakova po tekstu i tokenu i slično. Slikom 3.5. usporedno su prikazani brojevi tokena po tekstu u korpusu

koji nije pretprocesiran (graf lijevo) i u korpusu nad kojim je obavljeno pretprocesiranje (graf desno).



Slika 3.5. Usporedba broja tokena u originalnim i pretprocesiranim tekstovima

3.1.3. Podjela podataka

Za svaki model klasifikatora provedena je jednaka podjela skupa podataka na skup za učenje (engl. *training set*), skup za validaciju (engl. *validation set*) te skup za ocijenjivanje (engl. *testing set*). Podjela je provedena tako da skup za učenje sadrži 60% podataka iz početnog skupa podataka, dok skup za validaciju i ocijenjivanje sadrže svaki po 20% podataka iz početnog skupa. Podatci su se dijelili koristeći stratificiranu podjelu podataka. Ona osigurava da je svaka klasa u podijeljenim skupovima zastupljena u istom postotku kao i u originalnom skupu. U našem slučaju originalni skup podataka sadrži jednak broj primjera za svaku klasu tako da će isto vrijediti i za podijeljene skupove. U točnim brojkama, skup podataka za učenje sadrži 90 000 primjera, a skupovi podataka za validaciju i ispitivanje sadrže svaki po 30 000 primjera, odnosno tekstova.

3.2. Opis korištenih knjižnica

Za programsko ostvarenje klasifikatora namijenjenih prepoznavanju teksta generiranog velikim jezičnim modelom koristio se programski jezik Python i njegove dostupne knjižnice (engl. *libraries*). U programiranju, knjižnice su skupovi prethodno kompajliranog koda koje se mogu koristiti pri svaranju novih programskih rješenja. Mogu sadržavati prethodno kompajlirani kod, dokumentaciju, konfiguracijske podatke, klase i slično. U

programskom jeziku Python, knjižnice su kolekcije međusobno povezanih modula (engl. *modules*). Moduli su dokumenti koji mogu sadržavati izvršive izraze i definicije funkcija.

Python-ov standardni skup knjižnica (engl. *Python Standard Library*) sadrži već ugrađene module i knjižnice koje nije potrebno dodatno dohvaćati ili spremati. Od knjižnica sadržanih u Python-ovom skupu standardnih knjižnica korištene su *re* i *string* za manipuliranje s regularnim izrazima i tipovima podataka String, *collections*, *typing*, *logging*, *os*, *sys* i *time* za dokumentiranje, strukturiranje, ispisvanje i općenito pisanje programskog koda.

Od ostalih knjižnica koje je potrebno posebno instalirati korištene su knjižnice *Matplotlib* i *Seaborn* za vizualizacije grafova, *WordCloud* za vizualizacije teksta, *Pyplot* i *Graphviz* za vizualizaciju arhitektura modela.

Knjižnica *pandas* koristi se za rukovanje sa skupovima podataka i funkcijama vezanim uz manipulaciju i obradu skupova podataka. Uz nju, koristila se i knjižnica *NumPy* koja se koristi za rukovanje s poljima podataka i matricama.

Za pretprocesiranje tekstualnih podataka koristila se knjižnica *nltk* (engl. *Natural Language Toolkit*) koja sadrži funkcije potrebne za obradu prirodnog jezika. Za obradu tekstualnih podataka koristile su se i ranije spomenute knjižnice *re* i *string*.

Knjižnica *scikit-learn* koristila se za podjelu podataka na skupove za učenje, validaciju i ispitivanje te za računanje metrika za vrednovanje modela. Ova knjižnica općenito sadrži brojne funkcije i klase korisne za strojno učenje.

Za modeliranje koristile su se knjižnice *TensorFlow* i *transformers*. Knjižnica *TensorFlow* općenito je namijenjena za korištenje u području strojnog učenja. Može se koristiti za razne zadatke, ali najveći fokus stavljen je na učenje i rad s dubokim neuronskim mrežama i modelima zasnovanima na istima. S njezinog repozitorija *TensorFlow Hub* preuzeti su neki od prethodno treniranih modela korištenih za programsko ostvarenje klasifikatora u ovom radu. Knjižnicu *transformers* održava i razvija tvrtka *Hugging Face, Inc.* koja se bavi razvojem i korištenjem raznih modela strojnog učenja. *transformers* knjižnica također sadrži pred-trenirane modele korištene za razvoj programski ostvarenih klasifikatora u ovom radu.

4. Opis programski ostvarenih klasifikatora namijenjenih za prepoznavanje teksta generiranog velikim jezičnim modelom

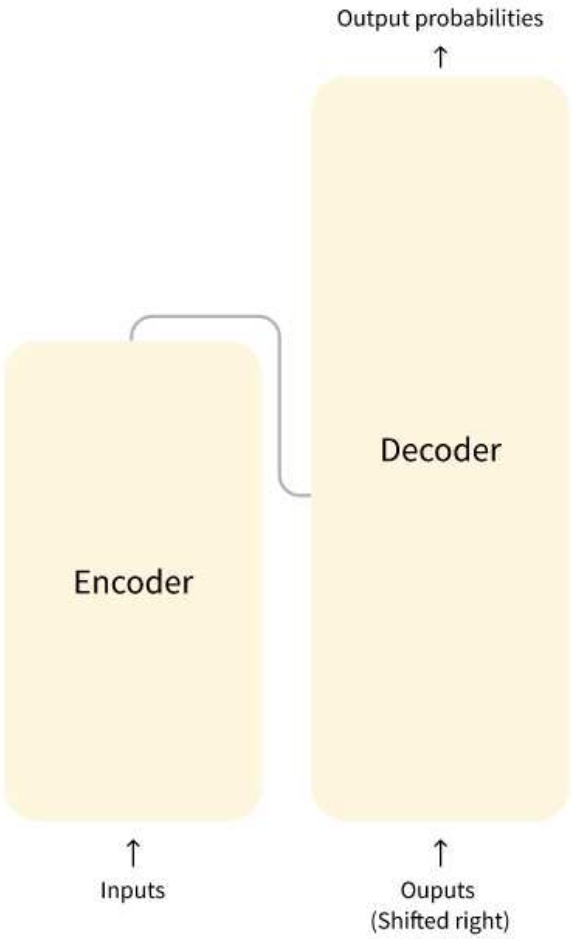
U ovom poglavlju detaljno će se opisati klasifikatori BERT, RoBERTa i DistilBERT korišteni za programsko ostvarenje prepoznavanja teksta generiranog velikim jezičnim modelom te njihova arhitektura i proces treniranja. Opisat će se i općenito rad modela zasnovanih na transformerima u području obrade prirodnog jezika.

4.1. Modeli zasnovani na transformerima

Modeli zasnovani na transformerima predstavljaju trenutno najnovije i najpreciznije arhitekture jezičnih modela u području obrade prirodnog jezika. Takvi modeli su prethodno naučeni (ili pred-trenirani (engl. *pretrained*)) nad velikom količinom podataka. Pred-treniranje je proces učenja modela iz nule, bez da model ima ikakvo prethodno znanje. Nakon pred-treniranja, jezični modeli zasnovani na transformerima mogu se koristiti za specifične zadatke. Da bi se naučili raditi predikcije za neki specifičan zadatak, moraju se ponovno trenirati nad novim podatcima koji su relevantni za taj zadatak. Tada proces učenja ne kreće od nule jer model već posjeduje neko znanje iz faze pred-treniranja. Proces treniranja pred-treniranog modela za specifični zadatak naziva se *fine tuning*.

Jedna od glavnih prednosti *fine tuning*-a je da je za postizanje dobrih rezultata za specifičan zadatak potrebno puno manje podataka jer su modeli već pred-trenirani nad velikim količinama općenitih podataka u obliku teksta. S treniranjem modela s manje podataka štedi se i na vremenu i resursima pri modeliranju [12].

Općenito, arhitektura modela zasnovanih na transformerima sastoji se od enkodera i dekodera, a prikazana je slikom 4.1. Enkoderi računaju vektorske reprezentacije ulaznih tokena, pritom uzimajući u obzir kontekst tih tokena. To znači da grade vektorske reprezentacije ulaznih tokena ne samo na osnovi njihovog semantičkog značenja, već i značenja tokena koji se nalaze u njihovoj neposrednoj blizini. Drugim riječima, enkoderi na ulaz primaju nizove tokena, a na izlaz vraćaju nizove vektora. Dekoderi primaju te nizove vektora kao ulazne podatke i kao izlaz vraćaju nizove tokena. Enkoderi i dekoderi sastoje se od takozvanih slojeva pažnje (engl. *attention layers*). Mehanizmi pažnje koji se koriste u tim slojevima sposobni su odrediti dugoročne povezanosti između riječi u rečenici. Otkriće ovih mehanizama označilo je prekretnicu u području obrade prirodnog jezika, a predstavljeni su u [1]. Slojevi pažnje funkcioniraju tako da računaju težine za svaki par riječi u rečenici. Te težine predstavljaju povezanost tih dviju riječi. Nakon računanja težina one se koriste za računanje težinske sume ulaznih vektora riječi [13].



Slika 4.1. Općenita arhitektura modela zasnovanih na transformerima [12]

Modeli zasnovani na transformerima ne moraju nužno koristiti i enkoder i dekoder. Na osnovi arhitekture, možemo ih podijeliti na modele koji koriste samo enkodere (engl. *encoder-only models*), modele koji koriste samo dekodere (engl. *decoder-only models*) te modele koji koriste i enkodere i dekodere (engl.*encoder-decoder models or sequence-to-sequence models*).

U ovom radu koristili su se modeli iz obitelji BERT koji koriste samo enkodere. Modeli koji koriste samo enkodere općenito su pogodni za probleme klasifikacije gdje je potrebno razumijeti ulazne tekstove i povezanosti riječi u njima. Slabiji su od modela koji koriste i enkodere i dekodere, ali su brži i manje iscrpni za treniranje [13].

Modeli koji koriste samo dekodere koriste se za zadatke u području obrade prirodnog jezika gdje je potrebno generirati neki tekst.

Modeli koji koriste i enkodere i dekodere koriste se za zadatke u području obrade prirodnog jezika gdje koji zahtijevaju ulaz u obliku teksta poput prevođenja teksta na neki drugi jezik ili slično. Modeli koji koriste dekodere su upravo oni koji su sposobni generirati tekstove koji se ovim radom pokušavaju prepoznati.

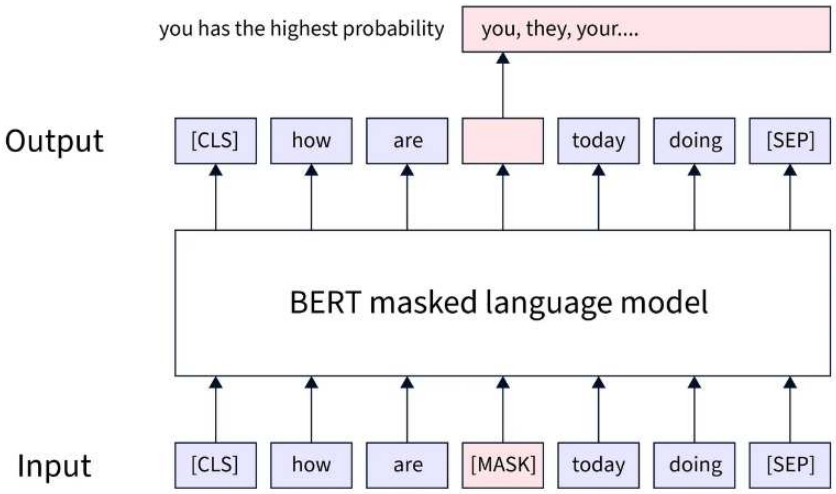
4.2. Klasifikator BERT

BERT (engl. *Bidirectional Encoder Representations from Transformers*) je model dubokog učenja zasnovan na transformerima. Transformeri povezuju izlazne podatke s ulaznim podatcima i njihovim težinama kako bi odredili njihovu razinu povezanosti. Ovakav mehanizam naziva se mehanizmom pažnje, a predstavljen je u [1]. Razvoj mehanizama koji koriste pažnju i razvoj modela BERT bile su neke od prekretnica u području obrade prirodnog jezika. Nakon što je predstavljen model BERT u [2], razvijeni su i mnogi njemu srodni modeli koji prate njegovu arhitekturu ili su inspirirani njome.

Modeli BERT prethodno su trenirani nad velikom količinom podataka pa je za njihovo korištenje za specifične probleme potrebno samo trenirati već postojeći model da bi se odabrali optimalni hiperparametri (engl. *fine tuning*). U svojoj arhitekturi, modeli BERT koriste samo enkodere, kao što je spomenuto u 4.1. Dijele se na dva tipa, *BERT base* i *BERT large*, ovisno o broju slojeva enkodera, slojeva pažnje i skrivenih vektora. Proces pred-treniranja modela BERT zasnovan je na dvije glavne komponente: korištenje ma-

skiranih jezičnih modela (engl. *Masked Language Models - MLMs*) i problem predviđanja sljedeće rečenice (engl. *next sentence prediction*).

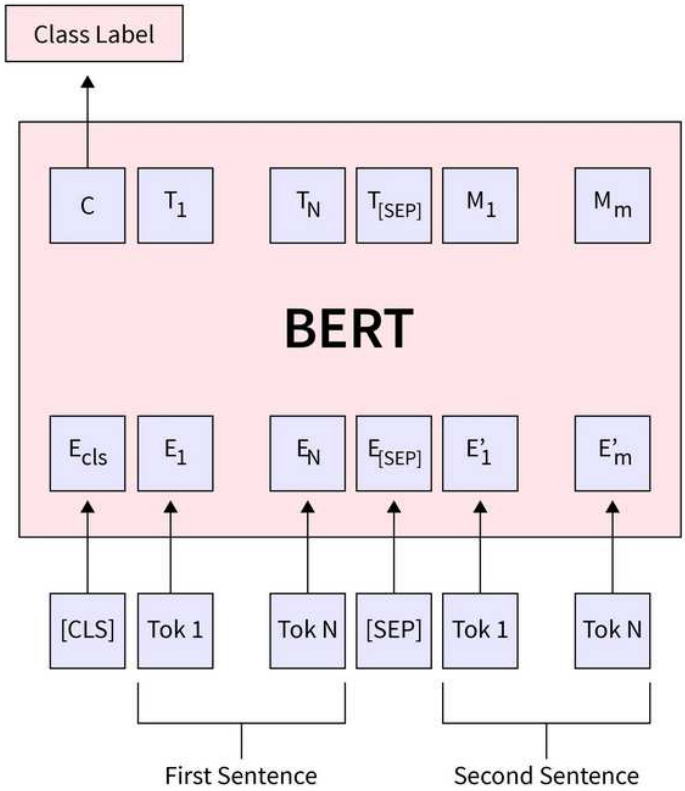
Jedna od prednosti modela BERT je njegova dvosmjernost (engl. *bidirectionality*) koja je postignuta koristeći maskirane jezične modele (engl. *Masked Language Models - MLMs*). Maskirani jezični modeli funkcioniraju tako da nasumično maskiraju neke od ulaznih tokena, odnosno riječi u tekstu. Cilj je da se ti tokeni onda prepoznaju samo na osnovu njihovog konteksta, odnosno na osnovu tokena koji se nalaze neposredno prije ili neposredno poslije maskiranog tokena. Za razliku od treniranja jezičnih modela s lijeva na desno (ili s desna na lijevo), koje je bilo uobičajeno dok se nije pojavio BERT, korištenjem pristupa s maskiranim jezičnim modelima omogućena je fuzija lijevog i desnog konteksta što rezultira s dvosmjernošću [2]. Primjer ulaznih i izlaznih podataka za maskirani jezični model prikazan je slikom 4.2. *[CLS]* je poseban simbol koji se dodaje na početak svakog ulaznog niza podataka, a *[SEP]* je poseban simbol koji služi za odvajanje (odgovora ili pitanja) [2].



Slika 4.2. Primjer rada maskiranog jezičnog modela [14]

Uz maskirane jezične modele, BERT koristi i zadatak predviđanja sljedeće rečenice (engl. *next sentence prediction*). Ovaj zadatak je uveden jer je za većinu problema obrade prirodnog jezika potrebno razumijeti kontekst ne samo u vidu susjednih riječi, već i susjednih rečenica. Zbog toga se u pred-treniranju BERT-a uvodi klasifikacija sljedeće rečenice. Radi se o binarnoj klasifikaciji gdje se za neki par rečenica A i B predviđa slijedi li rečenica B rečenici A ili je ona nasumično uzeta rečenica iz korpusa. Primjer predviđanja

sljedeće rečenice prikazan je slikom 4.3.



Slika 4.3. Primjer predviđanja sljedeće rečenice [15]

Neke od prednosti BERT-a su da je pogodan za specifičnije zadatke jer je pred-treniran na širokom skupu podataka [16]. Relativno je jednostavan za korištenje jer je pred-treniran i samo je potrebno podesiti njegove parametre (engl. *fine tuning*) za rješavanje potrebnog zadatka. Pokazao je znatno veću točnost u usporedbi s modelima poput neuronskih mreža koji su mu prethodili za probleme obrade prirodnog jezika.

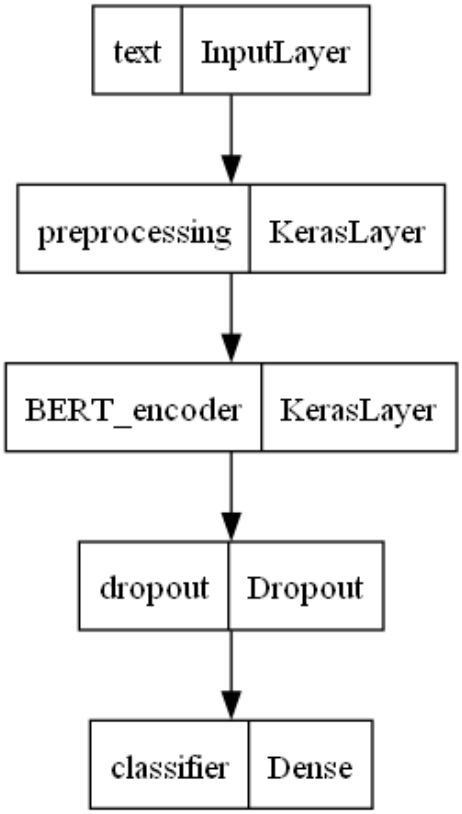
Ipak, BERT ima neka ograničenja koja su se pokušala ispraviti razvojem novih modela poput RoBERTa, DistilBERT-a i sličnih koji prate arhitekturu originalnog BERT-a. Neke od glavnih mana BERT modela su njegova veličina i računalna složenost koji izuskuju mnogo resursa prilikom njegovog treniranja.

4.2.1. Arhitektura programski ostvarenog klasifikatora BERT

Pri programskom ostvarenju klasifikatora BERT nije potrebna dodatna priprema podataka osim osnovnog pretprocesiranja opisanog u poglavlju 3.1.2. i podjele podataka opisane u poglavlju 3.1.3. BERT ima svoj pred-trenirani model za pretprocesiranje podataka

koji obrađuje podatke u sloju za pretprocesiranje.

Prije podešavanja parametara za problem koji se rješava, potrebno je odabrati i dohvatiti pred-trenirani model enkodera BERT iz repozitorija *TensorFlow Hub*. Postoji više pred-treniranih verzija enkodera BERT koji se razlikuju u količini parametara, razini preciznosti i slično. Model koji se koristio u ovom radu je *bert_en_uncased_L-12_H-768_A-12/3*. To je jedan od modela koji su razvili originalni autori rada u kojem je predstavljen BERT. Uz njega, potrebno je dohvatiti i ranije spomenuti model za pretprocesiranje koji mora odgovarati modelu enkodera. U ovom slučaju to je *bert_en_uncased_preprocess/3*. Kao što je prethodno objašnjeno u poglavlju 4.1., učitane pred-trenirane modele potrebno je naučiti procesom *fine tuning* za specifičan zadatak klasifikacije teksta generiranog velikim jezičnim modelom. Slikom 4.4. prikazana je arhitektura klasifikatora BERT razvijenog za prepoznavanje teksta generiranog velikim jezičnim modelom.



Slika 4.4. Arhitektura klasifikatora BERT

Klasifikator se sastoji od ulaznog sloja (engl. *input layer*) kojim se definira oblik i vrsta ulaznih podataka. Nakon ulaznog sloja slijedi Keras sloj za pretprocesiranje kojem se na ulaz predaju tekstovi. Ulazni tekstovi pretprocesiraju se Keras slojem, odnosno učitanim BERT modelom *bert_en_uncased_preprocess/3* za pretprocesiranje podataka. Pretproce-

sirani tekst se nakon toga predaje na ulaz sljedećeg Keras sloja, odnosno učitanog BERT enkodera *bert_en_uncased_L-12_H-768_A-12/3*. Izlazni podatci enkodera se zatim predaju tzv. *dropout* sloju koji nasumično postavlja te vrijednosti na nulu s frekvencijom *rate* koja je zadana kao hiperparametar. Ovaj sloj koristi se da bi se smanjila prenaučenost modela. Posljednji sloj je tzv. *gusti* (engl. *dense*) sloj koji primijenjuje aktivacijsku funkciju na izlazne podatke. Najbolji rezultati dobiveni su kada se pri definiranju gustog sloja nije navela aktivacijska funkcija, odnosno korištena je linearna kombinacija izlaza.

Za funkciju gubitka korištena je binarna unakrsna entropija (engl. *binary cross-entropy*) definirana formulom 4.1 gdje je *y* oznaka ciljne klase (je li tekst generiran ili nije), a vjerojatnost *p(y)* je predviđena vjerojatnost da je tekst generiran za svih N primjera. Za mjerenje empirijske pogreške korištena je metrika binarne točnosti. Točnost je definirana formulom 5.1, a binarna je jer postoje samo dvije klase, odnosno oznake.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{4.1}$$

Algoritam koji se koristio za optimizaciju je Adam. Adam je algoritam za stohastičku optimizaciju zasnovan na gradijentu [17]. Optimizator Adam naučen je s parametrom stope učenja od 2e-5. Klasifikator BERT je treniran u 3 epohe s veličinom serije (engl. *batch size*) 16.

4.3. Klasifikator RoBERTa

RoBERTa (engl. *Robustly Optimized BERT Pretraining Approach*) je poboljšana verzija modela BERT predstavljena s [18]. Autori modela RoBERTa predložili su nove izmjene u procesu treniranja modela BERT kako bi poboljšali rezultate i performanse modela. Uveli su promjene poput dužeg treniranja nad više podataka te kroz veće serije (engl. *batches*). Nadalje, shvatili su da uklanjanjem zadatka predviđanja sljedeće rečenice iz procesa pred-treniranja mogu poboljšati točnost modela. Trenirali su model nad dužim nizovima tokena te su uveli dinamičko mijenjanje uzorka maskiranja koji se primjenjuje tijekom treniranja modela [18]. Kod treniranja modela BERT tokeni ulaznih podataka maskirali su se na isti, statički, način kroz cijelo treniranje, odnosno uvijek bi se ma-

skirali tokeni na istim mjestima u nizu. Kod treniranja modela RoBERTa, primijenjeno je dinamičko maskiranje gdje se za svaki ulazni niz mijenja uzorak koji se maskira, tj. maskiraju se različiti tokeni u svakom koraku, što rezultira s time da model radi s raznolikijim podatcima i može bolje razumijeti kontekst. Također, model RoBERTa je trenirana sa znatno većim skupom tekstualnih podataka nego model BERT. Zbog toga je više vjerojatno da će model RoBERTa biti u mogućnosti prepoznati maskirane tokene i specifične riječi [19].

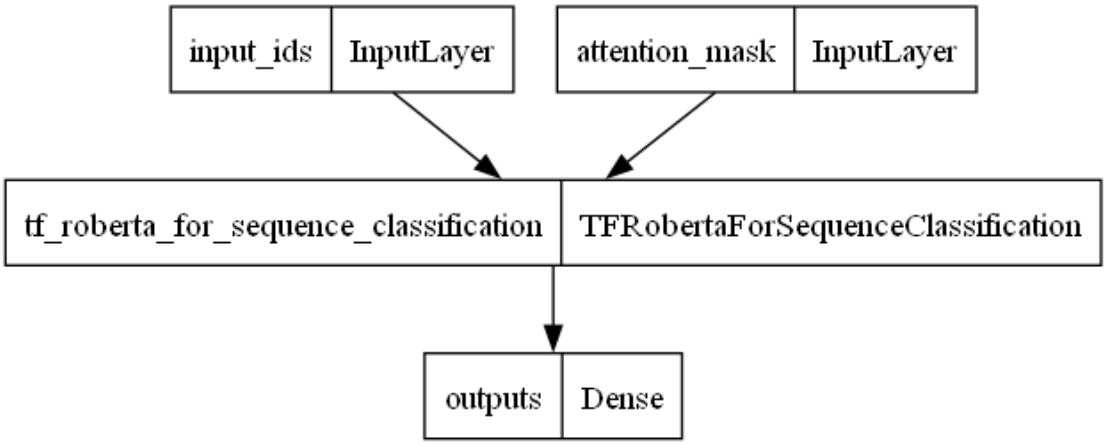
Kao što i samo ime modela RoBERTa kaže, on optimizira proces pred-treniranja modela BERT. To rezultira većom razinom preciznosti pri rješavanju problema obrade prirodnog jezika, ali i većim računalnim troškovima [20]. Jedna od većih prednosti modela RoBERTa je što je treniran nad tekstovima na više jezika pa se može koristiti i za multilingvističke probleme.

4.3.1. Arhitektura programski ostvarenog klasifikatora RoBERTa

Slično kao i kod modela BERT, model RoBERTa ima prethodno trenirani model tokenizatora za pretprocesiranje ulaznih podataka i prethodno trenirani model klasifikatora, kojem će se podesiti parametri, koje je potrebno učitati. Odabrani su modeli imena *roberta-base* predstavljeni u originalnom članku [18].

Arhitektura klasifikatora RoBERTa prikazana slikom 4.5. jednostavnija je od prethodno razvijenog klasifikatora BERT. Pri treniranju različitih kombinacija slojeva modela RoBERTa, pokazalo se da dodavanjem dodatnih slojeva dolazi do prenaučenosti i lošijih rezultata modela za problem prepoznavanja teksta generiranog velikim jezičnim modelom. Zbog toga se klasifikator sastoji od dvaju ulaznih slojeva (engl. *input layers*), sloja za klasifikaciju nizova i gustog (engl. *dense*) sloja.

Prije opisivanja slojeva arhitekture klasifikatora RoBERTa, potrebno je opisati i proces pretprocesiranja podataka koji obavlja tokenizator *roberta-base*. Klasifikator RoBERTa na ulaz prima nizove tokena, odnosno tekstove, jednake duljine pa je potrebno unaprijed odrediti maksimalnu duljinu niza tokena koje će klasifikator primiti na ulaz. Prethodno trenirani tokenizator *roberta-base* preoblikuje podatke na način da im skrati duljinu na predodređenu maksimalnu duljinu tokena ako je to potrebno ili, ako je duljina niza manja od maksimalne duljine tokena, dodaje takozvani *padding* kojim produljuje duljinu

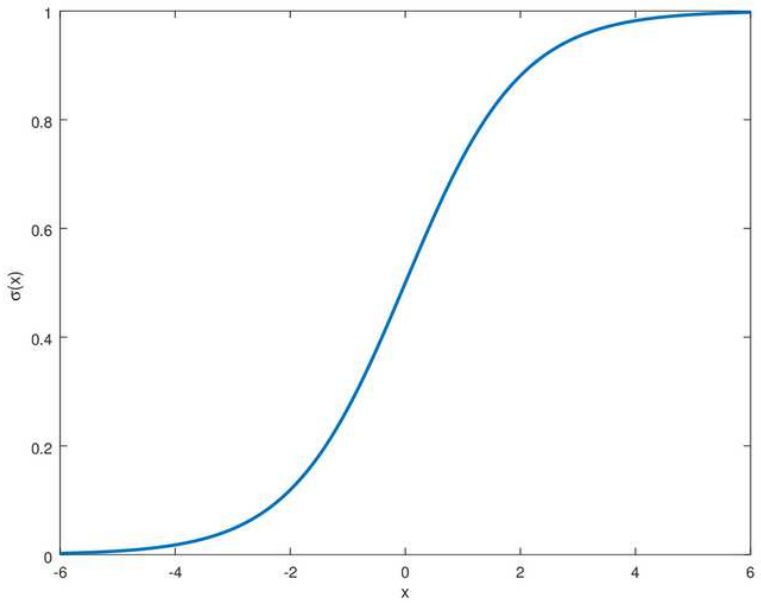


Slika 4.5. Arhitektura klasifikatora RoBERTa

niza do zadane maksimalne duljine. Maksimalne duljine nizova tokena uobičajeno je postaviti na neku potenciju broja 2. Analizom desnog grafa na slici 3.5. odabrana je maksimalna duljina teksta od 128 tokena. Nakon obrade podataka, tokenizator vraća dva polja vektora s elementima vrste *Tensor* iz knjižnice Tensorflow. Prvo polje sadrži identifikatore tokena za svaki niz podataka, a drugo polje sadrži maske tokena, tzv. *attention mask*. Maska niza tokena je polje duljine ulaznog niza tokena (predodređena veličina je 128) s vrijednostima nula i jedan gdje vrijednost nula označava da se na taj token ne treba obraćati pažnja pri treniranju modela. Primjer tokena na koje klasifikator ne bi trebao obraćati pažnju i uzimati u obzir pri klasifikaciji su tokeni koji služe da bi popunili ulazni niz tokena na predefiniranu duljinu (engl. *padding*). Ova dva polja podataka, identifikatori i maske tokena, su podatci za dva ulazna sloja modela, sloj *input_ids* i sloj *attention_mask*, respektivno.

Sljedeći sloj modela je *roberta-base* klasifikator nizova koji na ulaz dobiva podatke o identifikatorima i maskama tokena. Posljednji sloj je gusti (engl. *dense*) sloj koji primijenjuje aktivacijsku funkciju na izlazne podatke. Aktivacijska funkcija koja se koristila je sigmoidna funkcija definirana formulom 4.2 i prikazana slikom 4.6. Sigmoidna funkcija često se koristi zbog svog svojstva derivabilnosti i činjenice da transformira ulazne vrijednosti tako da poprimaju vrijenosti iz otvorenog intervala od 0 do 1.

$$\sigma(\alpha) = \frac{1}{1 + exp(\alpha)} \tag{4.2}$$



Slika 4.6. Sigmoidna funkcija

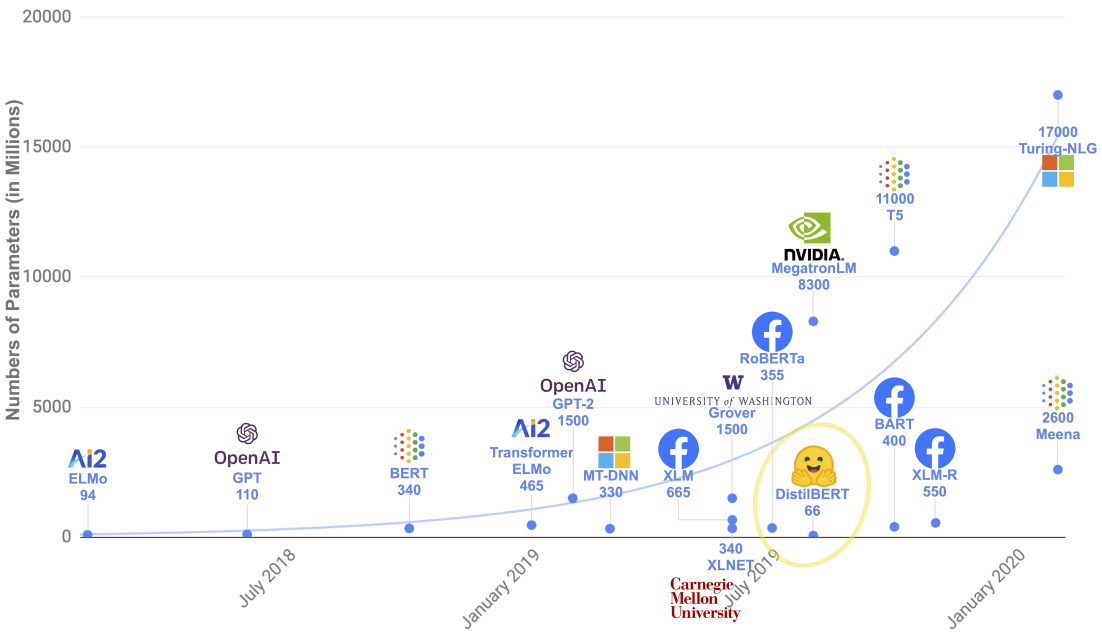
Za funkciju gubitka korištena je binarna unakrsna entropija (engl. *binary cross-entropy*) definirana formulom 4.1 gdje je *y* oznaka ciljne klase (je li tekst generiran ili nije), a vjerojatnost *p(y)* je predviđena vjerojatnost da je tekst generiran za svih N primjera. Za mjerenje empirijske pogreške korištena je metrika binarne točnosti. Točnost je definirana formulom 5.1, a binarna je jer postoje samo dvije klase, odnosno oznake. Za optimizaciju korišten je optimizator Adam [17] s parametrom stope učenja od 1e-5. Klasifikator RoBERTa treniran je kroz 3 epohe s veličinom serije (engl. *batch size*) 16.

4.4. Klasifikator DistilBERT

DistilBERT je model zasnovan na transformerima koji je nastao po uzoru na model BERT. Autori modela DistilBERT su destilacijom znanja u fazi pred-treniranja uspjeli smanjiti veličinu modela BERT za čak 40% istovremeno održavajući 97% njegovih sposobnosti razumijevanja jezika. Smanjenjem veličine originalnog modela BERT uspjeli su ubrzati novo stvoreni model za čak 60% [21].

Jedna od karakteristika modela DistilBERT koja ga izdvaja od većine novih jezičnih modela je što ima znatno manje parametara. Na slici 4.7. na osi ordinata, odnosno y osi, prikazani su brojevi parametara (u milijunima) jezičnih modela zasnovanih na transformerima. Os apscisa, odnosno x os, predstavlja trenutak u vremenu u kojem su objavljeni

modeli. Iz grafa je vidljivo da noviji modeli uglavnom prate trend porasta broja parametara, dok isto za model DistilBERT ne vrijedi. U kontekstu modela obitelji BERT, osnovni model BERT ima oko 340 milijuna parametara. Nakon njega objavljen je model RoBERTa koji ima nešto više parametara, oko 355 milijuna. Nakon toga slijedi model DistilBERT kojem broj parametara drastično pada na 66 milijuna. Autori DistilBERT bili su zabrinuti eksponencijalnim rastom broja parametara u novim jezičnim modelima pa su zbog ekoloških i praktičnih razloga odlučili drastično smanjiti resurse potrebne za rad njihovog modela. Tvrde da su njihovi DistilBERT modeli toliko mali da se mogu pokrenuti čak i na mobilnim uređajima [21].



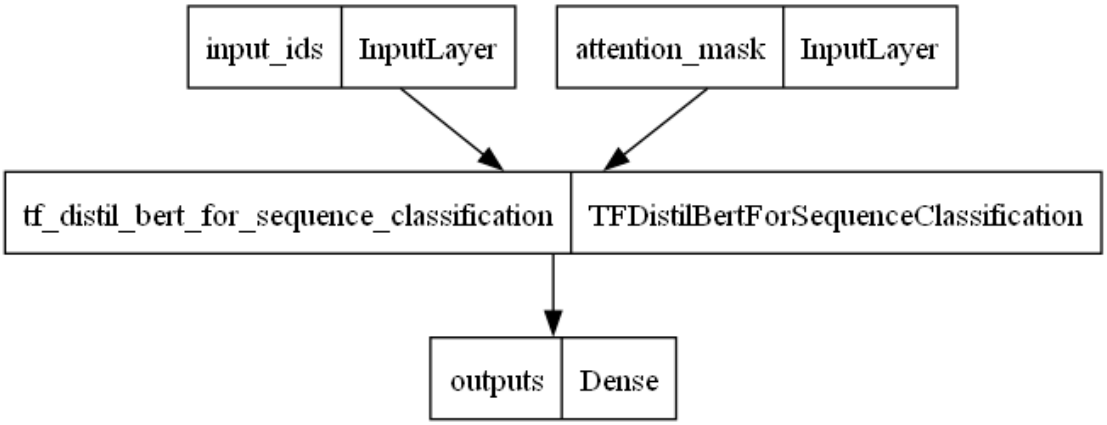
Slika 4.7. Broj parametara jezičnih modela kroz vrijeme [12]

Tehnika koja se koristila za reduciranje veličine DistilBERT modela naziva se destilacija znanja (engl. *knowledge distillation*). To je kompresijska tehnika u kojoj je kompaktni model - učenik, treniran na način da reproducira ponašanje većeg modela - učitelja ili ansambla modela. Ova tehnika uvedena je s [22], a generalizirana u [23]. U kontekstu modela DistilBERT, model DistilBERT je učenik, a model BERT, kao veći model čije ponašanje učenik DistilBERT pokušava ponoviti, je učitelj.

4.4.1. Arhitektura programski ostvarenog klasifikatora DistilBERT

Postupak treniranja modela i prethodne obrade podataka za klasifikator DistilBERT jednak je kao i za klasifikator RoBERTa opisan u poglavlju 4.3.1. Jedina razlika je što se za klasifikator DistilBERT, naravno, učitavaju tokenizator i klasifikator nizova s arhitekturom modela DistilBERT. Specifično, koristila se verzija tokenizatora i klasifikatora DistilBERT s nazivom *distilbert-base-uncased*, predstavljena u radu [21].

Arhitektura programski ostvarenog klasifikatora DistilBERT prikazana je slikom 4.8. Pri programskom ostvarenju klasifikatora DistilBERT, korišteni su isti tipovi slojeva s istim parametrima kao i kod klasifikatora RoBERTa da bi se modeli mogli pravilno usporediti jedan s drugime.



Slika 4.8. Arhitektura klasifikatora DistilBERT

Ulazni slojevi (engl. *input layers*) klasifikatora DistilBERT predstavljaju identifikatore te masku pažnje za ulazni niz tokena. Sljedeći sloj je klasifikator nizova *distilbert-base-uncased*, odnosno pred-trenirani DistilBERT model kojeg je sada potrebno dodatno trenirati za specijalizaciju nad problemom prepoznavanja teksta generiranog velikim jezičnim modelom. Klasifikator nizova *distilbert-base-uncased* na ulaz prima podatke koje je prethodno obradio tokenizator *distilbert-base-uncased*. Tokenizator *distilbert-base-uncased* obrađuje podatke na isti način kao i tokenizator *roberta-base*, opisan u poglavlju 4.3.1. Posljednji sloj je gusti (engl. *dense*) sloj koji na ulaz prima predikcije relevantnosti za svaku klasu koje za svaki primjer računa klasifikator u prethodnom sloju. S obzirom na to da se radi o binarnoj klasifikaciji, odnosno postoje dvije klase, gusti sloj će na ulaz dobiti dvije predviđene vjerojatnosti pripadanja određenog primjera svakoj klasi. Nad

tim vrijednostima gusti sloj će primijeniti aktivacijsku funkciju. Nakon toga će se dobiti konačan izlaz, a to je klasa za koju je klasifikator odredio da je više vjerojatna da joj taj primjer pripada. Aktivacijska funkcija koja se koristila je sigmoidna funkcija prikazana slikom 4.6. i definirana formulom 4.2

Pri programskom ostvarenju klasifikatora DistilBERT, za funkciju gubitka korištena je binarna unakrsna entropija (engl. *binary cross-entropy*) definirana formulom 4.1 gdje je y oznaka ciljne klase (je li tekst generiran ili nije), a vjerojatnost $p(y)$ je predviđena vjerojatnost da je tekst generiran za svih N primjera. Za mjerenje empirijske pogreške korištena je metrika binarne točnosti. Točnost je definirana formulom 5.1, a binarna je jer postoje samo dvije klase, odnosno oznake. Za optimizaciju korišten je optimizator Adam [17] s parametrom stope učenja od 1e-5. Klasifikator DistilBERT treniran je kroz 3 epohe s veličinom serije (engl. *batch size*) 16.

Proces treniranja velikih jezičnih modela vrlo je iscrpan, pogotovo ako se treniraju s velikom količinom podataka što je potrebno za dobivanje što boljih rezultata. Zato je bitno za istaknuti da je faza učenja modela DistilBERT trajala čak četiri puta manje nego faze učenja modela BERT i RoBERTa.

5. Opis metrika za vrednovanje uspješnosti prepoznavanja teksta generiranog velikim jezičnim modelom

U ovom poglavlju će se detaljno opisati metrike korištene za vrednovanje programski ostvarenih klasifikatora za problem prepoznavanja teksta generiranog velikim jezičnim modelom.

5.1. Matrica zabune

Matrica zabune (engl. *confusion matrix*) je kvadratna matrica koja prikazuje količinu ispravno i netočno predviđenih primjera. U slučaju binarne klasifikacije, koja je prisutna u problemu prepoznavanja teksta generiranog velikim jezičnim modelom, matrica zabune je dimenzija 2 x 2 i općenito je prikazana slikom 5.1.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Slika 5.1. Matrica zabune [24]

Na slici 5.1. retci matrice predstavljaju predviđene vrijednosti, a stupci stvarne vrijednosti. Postoje četiri elemenata matrice, odnosno četiri moguća slučaja: *stvarno pozitivni*

(engl. *true positives - TP*) gdje je predikcija jednaka 1 i stvarna oznaka je jednaka 1, *lažno pozitivni* (engl. *false positives - FP*) gdje je predikcija jednaka 1, a stvarna oznaka je jednaka 0, *lažno negativni* (engl. *false negatives - FN*) gdje je predikcija jednaka 0, a stvarna oznaka je jednaka 1 te *stvarno negativni* (engl. *true negatives - TN*) gdje je predikcija jednaka 0 i stvarna oznaka je jednaka 0.

Matrica zabune kao takva nije metrika, ali se očitavanjem njezinih elemenata mogu računati razne metrike objašnjene u nadolazećim poglavljima.

5.2. Točnost

Mjera točnosti (engl. *accuracy*) je udio ispravno klasificiranih primjera u skupu svih primjera. Računa se koristeći formulu 5.1

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

Problem s mjerom točnosti je što ako se primijeni na problem s neuravnoteženim skupom podataka može rezultirati nepouzdano visokim rezultatima. Ako bi neki klasifikator predviđao samo većinski zastupljenu klasu, rezultat točnosti bio bi vrlo visok, dok bi predikcije za manje zastupljenu klasu bile u potpunosti netočne, ali se to ne bi moglo iščitati iz mjere točnosti. Zato je uvijek pouzdano koristiti više metrika uz točnost.

5.3. Preciznost

Mjera preciznosti (engl. *precision*) definira se kao udio stvarno pozitivnih primjera u skupu svih primjera koje je klasifikator označio kao pozitivnima. Računa se koristeći formulu 5.2

$$P = \frac{TP}{TP + TN} \tag{5.2}$$

Cilj je da je vrijednost preciznosti što bliža (ili jednaka) 1.

5.4. Odziv

Mjera odziva (engl. *recall*) definira se kao udio stvarno pozitivnih primjera u skupu svih pozitivnih primjera. Računa se koristeći formulu 5.3

$$R = \frac{TP}{TP + FN} \tag{5.3}$$

Cilj je da je vrijednost odziva jednaka 1, odnosno da su svi pozitivni primjeri ispravno klasificirani. Alternativni naziv za odziv je stopa stvarnih pozitivnih vrijednosti (engl. *true positive rate*).

5.5. Mjera F1

Mjera F1 (engl. *F1-score*) definirana je kao harmonijska sredina mjera preciznosti i odziva. Preciznost i odziv su međusobno oprečne metrike, kada klasifikator ima visoku preciznost, imat će niži odziv i obrnuto. Zbog toga se mjera F1 često koristi da bi se dobila jedinstvena vrijednost koja predstavlja kvalitetu predikcija klasifikatora. Mjera F1 računa se koristeći formulu 5.4

$$F1 = \frac{2RP}{R + P} \tag{5.4}$$

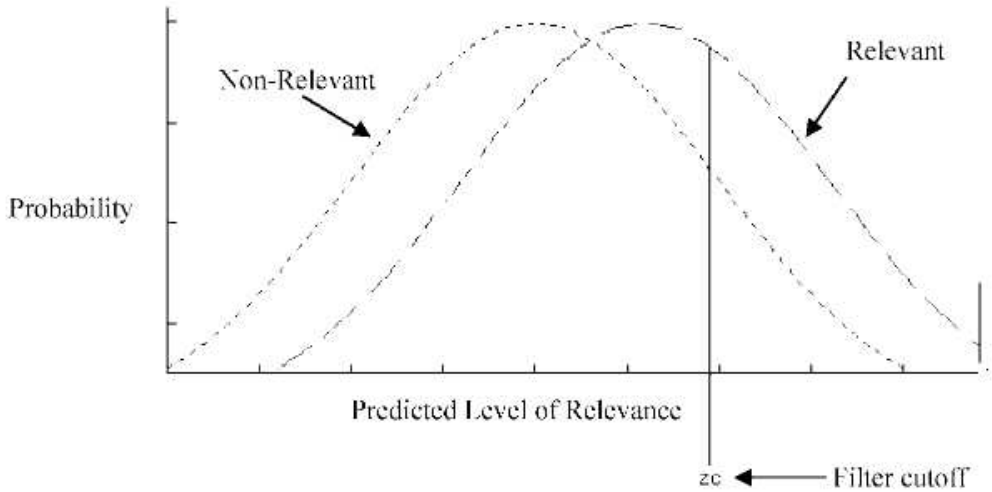
Kao i kod preciznosti i odziva, idealno je da je F1 vrijednost jednaka 1.

Općenito, mjera F-beta se računa koristeći formulu 5.5 Za vrijednosti parametra β manje od 1 naglašava se mjera preciznosti, a za vrijednosti parametra β veće od 1 naglašava se mjera odziva. Mjera F1 je jednaka mjeri F-beta s parametrom β jednakim 1 gdje se pridaje jednaka važnost mjeri odziva i preciznosti.

$$F_\beta = \frac{(1 + \beta^2)RP}{\beta^2 P + R} \tag{5.5}$$

5.6. ROC krivulja i površina ispod ROC krivulje

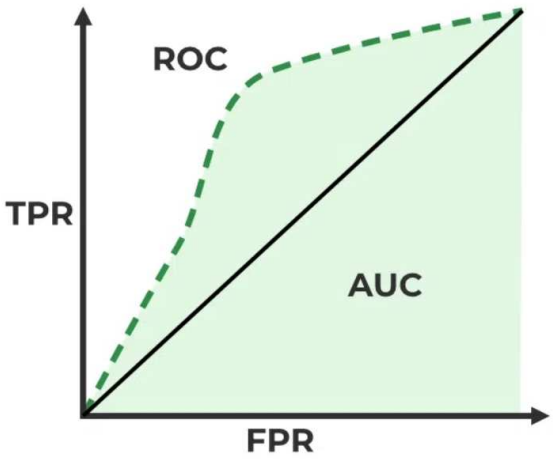
Metoda ROC (engl. *Receiver Operating Characteristic*) krivulja koristi se za vrednovanje sposobnosti diskriminiranja modela između različitih grupa [25]. ROC model radi pod pretpostavkom da će informacijski sustav svakom primjeru dodijeliti predikciju relevantnosti za svaku klasu. U slučaju binarne klasifikacije teksta klasifikator će svakom tekstu dodijeliti predikcije relevantnosti za obje klase. Uz ovu pretpostavku modela ROC, postojat će dvije distribucije prikazane slikom 5.2. Lijeva distribucija predstavlja vjerojatnost da će klasifikator dati predikciju razine važnosti (os x) klasi koja nije relevantna za taj primjer, odnosno klasi kojoj toj primjer ne pripada. Desna distribucija predstavlja vjerojatnosnu distribuciju za klasu primjera koji jesu relevantni, odnosno za klasu kojoj taj primjer stvarno pripada. Što su te dvije vjerojatnosne distribucije udaljenije jedna od druge, to je model bolji u diferenciranju između klasa [26].



Slika 5.2. Prikaz mogućih funkcija gustoće za relevantnu i ne relevantnu klasu [26]

ROC krivulja prikazuje odziv (engl. *recall*), odnosno stopu stvarnih pozitivnih vrijednosti (engl. *True Positive Rate - TPR*) kao funkciju stope lažno pozitivnih vrijednosti (engl. *False Positive Rate - FPR*). Stopa stvarnih pozitivnih vrijednosti definirana je formulom 5.3, a stopa lažno pozitivnih vrijednosti definirana je formulom 5.6 Površina ispod ROC krivulje (engl. *Area Under the Curve - AUC*) je mjera koja se koristi za ispitivanje koliko model dobro klasificira primjere. ROC krivulja i AUC prikazane su slikom 5.3.

$$FPR = \frac{FP}{TN + FP} \tag{5.6}$$



Slika 5.3. Prikaz ROC krivulje i površine ispod krivulje (AUC) [27]

Vrijednosti površine ispod krivulje kreću se u zatvorenom intervalu od 0 do 1. Idealan slučaj bi bio da je površina ispod krivulje jednaka 1, što bi značilo da model savršeno diskirminira između klasa. Tada bi ROC krivulja zatvarala pravi kut s osi y.

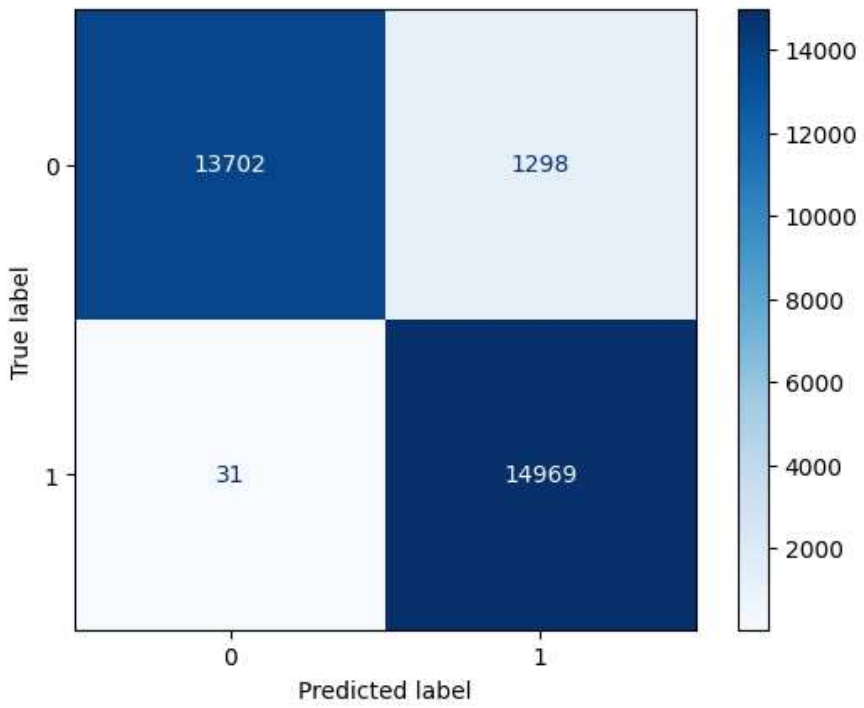
6. Rezultati i rasprava

U ovom poglavlju detaljno će se vrednovati sva tri programski ostvarena klasifikatora za problem prepoznavanja teksta generiranog velikim jezičnim modelom. Koristit će se metrike opisane u poglavlju 5. Klasifikatori će se vrednovati nad neviđenim podatcima za ispitivanje iz skupa podataka *„GPT Wiki Intro"* opisanim u poglavlju 3.1. te nad dodatnim podatcima za ispitivanje opisanim u poglavlju 3.1.1. Rezultati će se nakon toga usporediti i prodiskutirati.

6.1. Rezultati klasifikatora BERT

Nakon što je model BERT naučen, ispitan je na skupu za ispitivanje koji sadrži 30 tisuća primjera iz originalnog *„GPT Wiki Intro"* skupa podataka. Skup za ispitivanje sadrži jednak broj primjera iz obiju klasa, odnosno jednak broj tekstova koje je generirao veliki jezični model Curie i tekstova koje su napisali ljudi. Matrica zabune za predikcije klasifikatora BERT prikazana je slikom 6.1. Postoji 1298 lažno pozitivnih primjera (engl. *False Positive - FP*) i samo 31 lažno negativan (engl. *False Negative - FN*) primjer. Za specifičan problem prepoznavanja tekstova generiranih velikim jezičnim modelima bolji je slučaj kada je neki tekst pogrešno označen kao da je generiran, a zapravo ga je napisao čovjek (FP). Ako bi se osoba koja tvrdi da je autor tog teksta žalila na donesenu odluku, taj slučaj bi se raspravio i utvrdila bi se greška modela. U drugom slučaju u kojem model pogrešno označi tekst kao da ga je napisao čovjek, a zapravo ga je generirao neki veliki jezični model (FN), vrlo je vjerojatno da se dodatne provjere nebi radile i da bi taj tekst ostao neopažen. S obzirom na to da model BERT ima puno manje lažno negativnih primjera nego lažno pozitivnih, njegov rad je pogodan za ovaj specifičan problem.

Koristeći matricu zabune izračunate su mjere točnosti, odziva, preciznosti i F1 mjera. U tablici 6.1. prikazani su rezultati metrika za klasifikator BERT. S obzirom na to da su



Slika 6.1. Matrica zabune klasifikatora BERT

Metrika	Vrijednost
Točnost	0.956
Preciznost	0.920
Odziv	0.998
F1 mjera	0.957

Tablica 6.1. Rezultati vrednovanja klasifikatora BERT

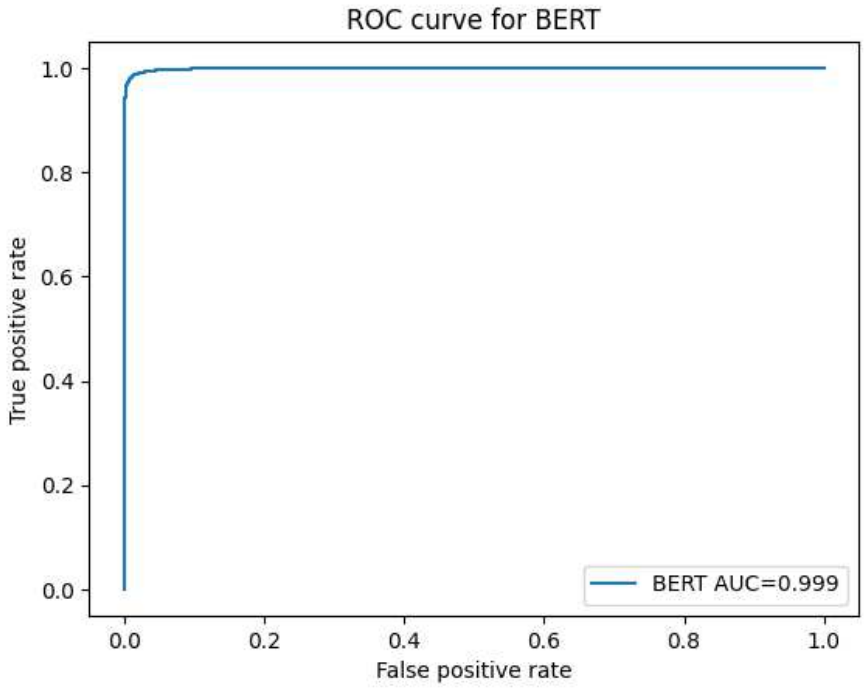
za slučaj prepoznavanja teksta generiranog velikim jezičnim modelom bitnije predikcije pozitivne klase (klase koja označava da je tekst generiran velikim jezičnim modelom), mjera odziva je vrlo bitna. Za klasifikator BERT ona je vrlo visoka što potvrđuje prethodna iščitavanja iz matrice zabune koja dovode do zaključka da je klasifikator BERT pogodan za ovaj problem.

Također, iste metrike izračunate su i za svaku klasu posebno, tzv. klasifikacijski izvještaj (engl. *classification report*). Rezultati su prikazani tablicom 6.2. Iz tablice je vidljivo da je odziv za pozitivnu klasu praktički stopostotan (radi se o zaokruženoj vrijednosti).

Dodatno, slikom 6.2. prikazana je ROC krivulja. Iz grafa je vidljivo da je kut koji krivulja zatvara s osi y blizu pravome kutu, što je idealan slučaj. Vrijednost površine ispod krivulje (AUC) iznosi 0.9990.

Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.91	0.95	15 000
1	0.92	1.00	0.96	15 000
prosjek	0.96	0.96	0.96	30 000

Tablica 6.2. Klasifikacijski izvještaj za klasifikator BERT



Slika 6.2. ROC krivulja za klasifikator BERT

6.1.1. Rezultati klasifikatora BERT nad dodatnim podatcima za ispitivanje

Klasifikator BERT dodatno je vrednovan nad dva skupa podataka opisana u poglavlju 3.1.1. Jedan od njih ispituje negativnu klasu, odnosno koliko dobro klasifikator klasificira tekstove koje je napisao čovjek. Taj skup podataka sadrži 1375 eseja koje su napisali učenici i 3 eseja koje je generirao neki veliki jezični model. Matrica zabune klasifikatora BERT za ovaj skup podataka prikazana je slikom 6.3. Koristeći matricu zabune izračunate su metrike preciznosti, odziva i F1 mjera za svaku klasu. Rezultati su prikazani tablicom 6.3. Neke metrike, poput mjere preciznosti, ovise o broju primjera iz drugih klasa. U slučaju nebalansiranih klasa (engl. *class imbalance*), odnosno kada jedna klasa ima znatno više primjera od druge, takve metrike imat će iskrivljene rezultate. Zato je u ovom slučaju, gdje postoji nebalansiranost klasa, bitno odabrati i promatrati ispravne metrike. Metrika koja ne uzima u obzir broj primjera iz drugih klasa je odziv tako da

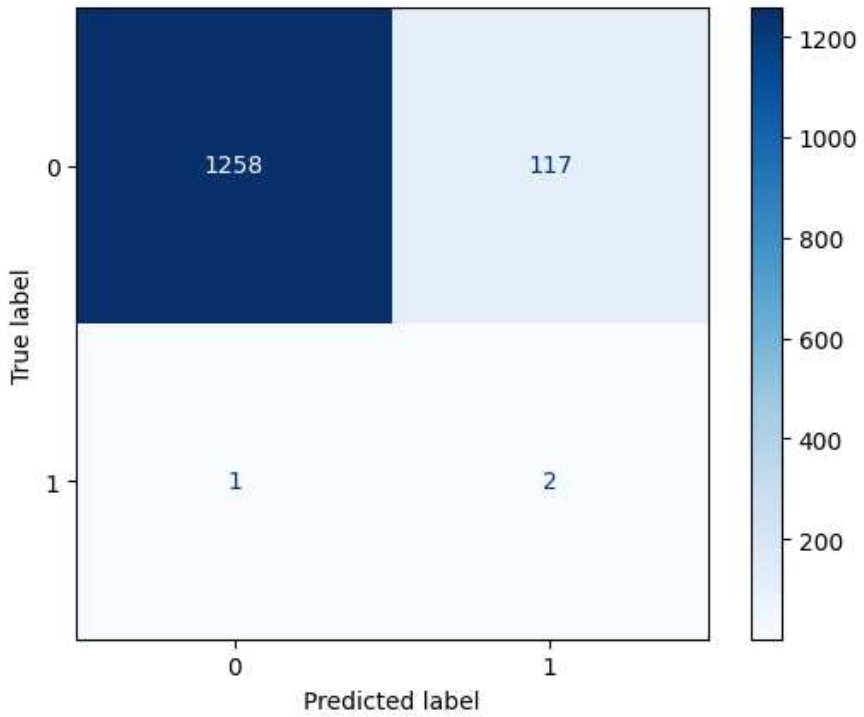
Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.91	0.96	1375
1	0.02	0.67	0.03	3
težinski prosjek	1.00	0.91	0.95	1378

Tablica 6.3. Klasifikacijski izvještaj za klasifikator BERT za skup podataka koji ispituje negativnu klasu

Klasa	**Odziv**	Broj primjera
1	0.82	3611

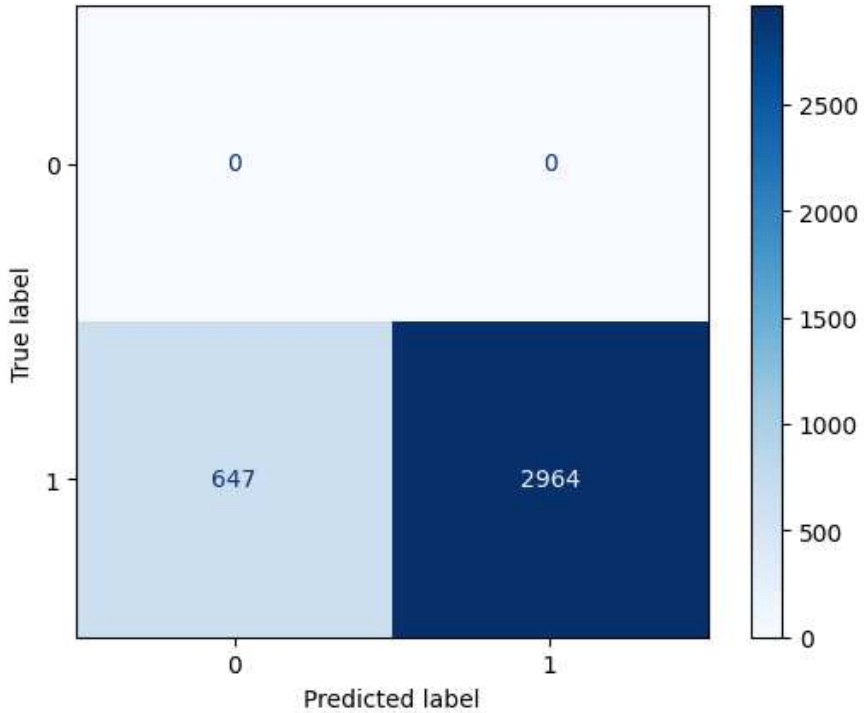
Tablica 6.4. Klasifikacijski izvještaj za klasifikator BERT za skup podataka koji ispituje pozitivnu klasu

je za ovaj skup podataka najispravnije promatrati tu metriku. Njezini rezultati ukazuju na to da klasifikator BERT daje dobre predikcije i za potpuno drukčije podatke od onih s kojima je treniran.



Slika 6.3. Matrica zabune klasifikatora BERT za skup podataka koji ispituje negativnu klasu

Drugi skup podataka za dodatno ispitivanje sastoji se od 3611 primjera tekstova koje su generirali veliki jezični modeli PaLM, Falcon i LLaMA. Matrica zabune klasifikatora BERT za ovaj skup podataka prikazana je slikom 6.4. Koristeći matricu zabune izračunate je metrika odziva i F1 mjera za pozitivnu klasu jer primjera iz negativne klase nema. Rezultati su prikazani tablicom 6.4.



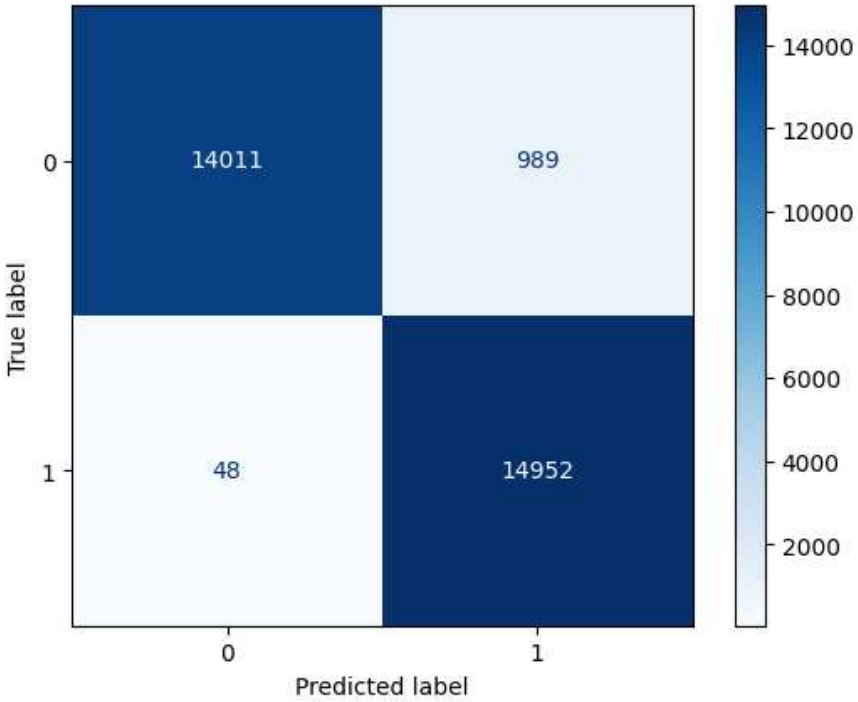
Slika 6.4. Matrica zabune klasifikatora BERT za skup podataka koji ispituje pozitivnu klasu

6.2. Rezultati klasifikatora RoBERTa

Nakon što je model RoBERTa naučen, vrednovan je sa skupom za ispitivanje koji sadrži 30 tisuća primjera iz originalnog *„GPT Wiki Intro”* skupa podataka, odnosno 20% primjera. Skup za ispitivanje sadrži jednak broj primjera iz obje klase. Matrica zabune klasifikatora RoBERTa prikazana je slikom 6.5. Postoji 989 primjera koji su klasificirani kao lažno pozitivni (FP) te 48 primjera koji su klasificirani kao lažno negativni (FN). Ove brojke ukazuju na to da klasifikator RoBERTa rijetko griješi, a i kada radi greške, radi ih znatno manje za pozitivnu klasu što je pogodno za problem prepoznavanja teksta generiranog velikim jezičnim modelom.

Koristeći informacije iz matrice zabune, izračunate su mjere odziva, točnosti, preciznosti i F1 mjera za primjere iz skupa za ispitivanje. Rezultati su prikazani tablicom 6.5. Rezultati su uglavnom visoki, pogotovo za mjeru odziva koja je bitna za problem prepoznavanja teksta generiranog velikim jezičnim modelom.

Nadalje, napravljen je klasifikacijski izvještaj (engl. *classification report*) za svaku klasu. Rezultati su prikazani tablicom 6.6. Uz vrlo mali broj lažno negativnih primjera, zaokruženi rezultat mjere odziva je idealnih 1.



Slika 6.5. Matrica zabune klasifikatora RoBERTa

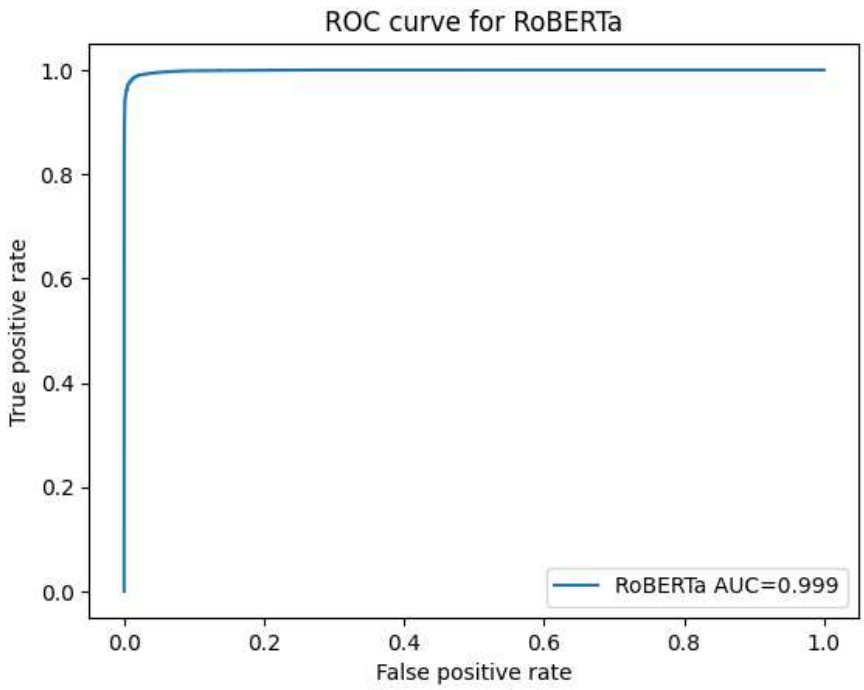
Metrika	**Vrijednost**
Točnost	0.965
Preciznost	0.938
Odziv	0.997
F1 mjera	0.966

Tablica 6.5. Rezultati vrednovanja klasifikatora RoBERTa

Slikom 6.6. prikazana je ROC krivulja za klasifikator RoBERTa. Iz grafa je vidljivo da ROC krivulja skoro zatvara pravi kut s osi y, što je idealan slučaj. Vrijednost površine ispod ROC krivulje (AUC) iznosi 0.9989.

Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.93	0.96	15 000
1	0.94	1.00	0.97	15 000
prosjek	0.97	0.97	0.97	30 000

Tablica 6.6. Klasifikacijski izvještaj za klasifikator RoBERTa



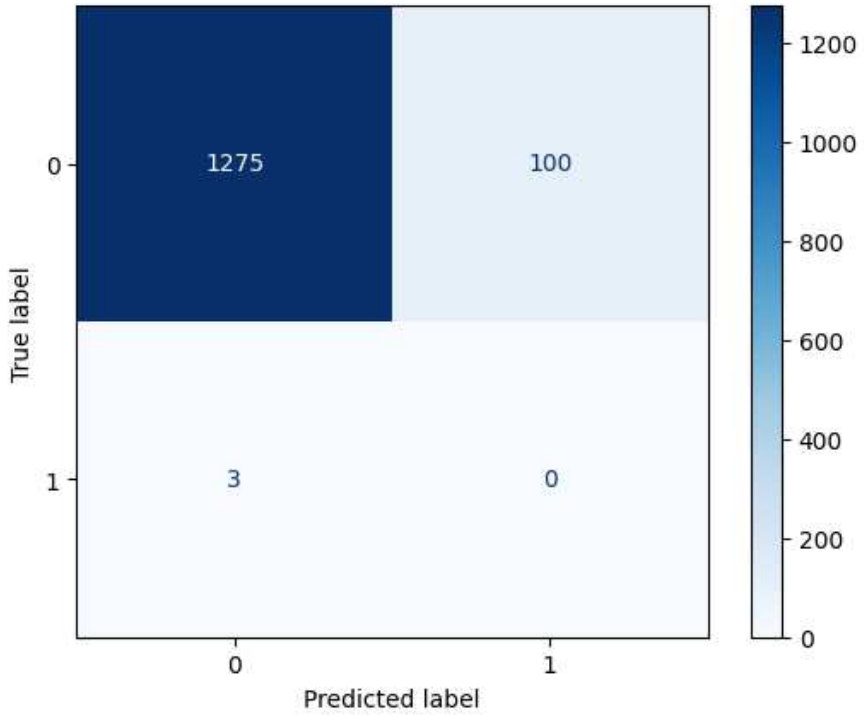
Slika 6.6. ROC krivulja za klasifikator RoBERTa

6.2.1. Rezultati klasifikatora RoBERTa nad dodatnim podatcima za ispitivanje

Rad klasifikatora RoBERTa dodatno je ispitan s dva skupa podataka koji imaju primjere za svaku klasu opisani u poglavlju 3.1.1. Prvi skup podataka koji sadrži eseje koje su napisali učenici služi za ispitivanje rada klasifikatora za negativnu klasu, tj. klasu primjera s oznakom 0 (*not generated*). Matrica zabune za predikcije klasifikatora RoBERTa nad ovim skupom podataka prikazana je slikom 6.7. Model je imao malo pogrešno klasificiranih negeneriranih tekstova, ali nije točno klasificirao niti jedan od 3 primjera za generirane tekstove, što je svakako manje bitno za ovaj skup podataka.

Korištenjem vrijednosti iz matrice zabune izračunate su mjere preciznosti, odziva i mjera F1 za svaku klasu. Ovi rezultati prikazani su tablicom 6.7. Sve mjere za pozitivnu klasu jednake su 0 jer klasifikator nije ispravno predvidio niti jedan primjer. S druge strane, mjera preciznosti za negativnu klasu rezultirala je idealnom vrijednošću zbog vrlo male količine predmeta negativne klase. Metrika koja u ovom slučaju daje pouzdanije rezultate je odziv koji pokazuje visoku razinu točnosti, ali i realno stanje.

Nadalje, drugi skup podataka koji služi za ispitivanje rada klasifikatora nad pozitivnom klasom sastoji se od 3611 primjera. Matrica zabune klasifikatora RoBERTa za ovaj



Slika 6.7. Matrica zabune klasifikatora RoBERTa za skup podataka koji ispituje negativnu klasu

Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.93	0.96	1375
1	0.00	0.00	0.00	3
težinski prosjek	1.00	0.93	0.96	1378

Tablica 6.7. Klasifikacijski izvještaj za klasifikator RoBERTa za skup podataka koji ispituje negativnu klasu

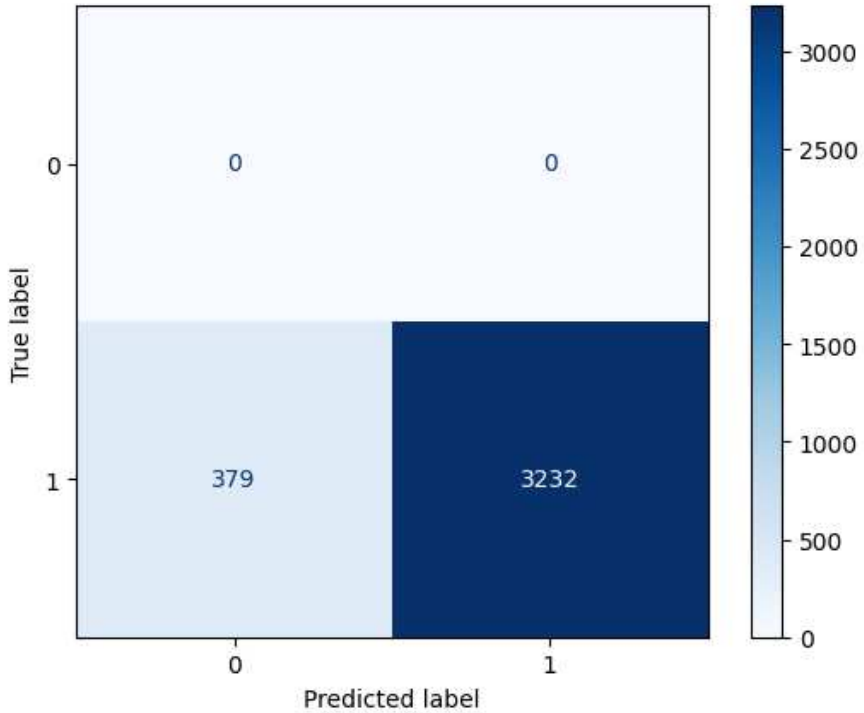
skup podataka prikazana je slikom 6.8. Iz slike se vidi da je klasifikator RoBERTa vrlo malo primjera pogrešno klasificirao. Koristeći ove vrijednosti izračunata je mjera odziva za pozitivnu klasu prikazana tablicom 6.8. Klasifikator RoBERTa pokazuje vrlo dobre rezultate i na skupu za ispitivanje i na dodatnim skupovima podataka.

6.3. Rezultati klasifikatora DistilBERT

Klasifikator DistilBERT je vrednovan nad skupom podataka za ispitivanje koji sadrži 30 tisuća primjera od originalnog skupa podataka koji je bio podijeljen prije procesa učenja.

Klasa	**Odziv**	Broj primjera
1	0.90	3611

Tablica 6.8. Klasifikacijski izvještaj za klasifikator RoBERTa za skup podataka koji ispituje pozitivnu klasu



Slika 6.8. Matrica zabune klasifikatora RoBERTa za skup podataka koji ispituje pozitivnu klasu

Metrika	Vrijednost
Točnost	0.963
Preciznost	0.934
Odziv	0.997
F1 mjera	0.964

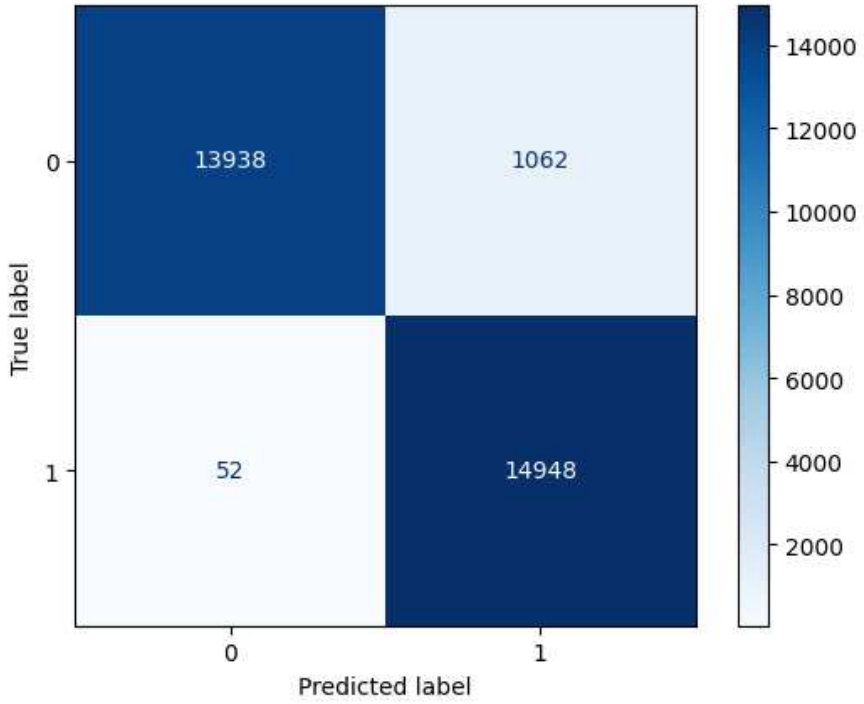
Tablica 6.9. Rezultati vrednovanja klasifikatora DistilBERT

Skup podataka za ispitivanje sadrži jednak broj primjera označenih kao generirani i označenih kao da nisu generirani velikim jezičnim modelom. Matrica zabune klasifikatora DistilBERT prikazana je slikom 6.9.

Uz pomoć vrijednosti elemenata matrice zabune izračunate su mjere točnosti, odziva, preciznosti i F1 mjere za klasifikator DistilBERT. Rezultati ovih mjera prikazani su tablicom 6.9.

Izračunate su i metrike odziva, preciznosti i F1 mjera za svaku od klasa. Rezultati ovog klasifikacijskog izvještaja prikazani su tablicom 6.10.

ROC krivulja koja govori koliko model dobro diskriminira između klasa prikazana je slikom 6.10. Krivulja je vrlo blizu zatvaranja pravog kuta s y osi što ju čini vrlo blizu idealnog slučaja. Površina ispod krivulje (AUC) iznosi 0.9987.



Slika 6.9. Matrica zabune klasifikatora DistilBERT

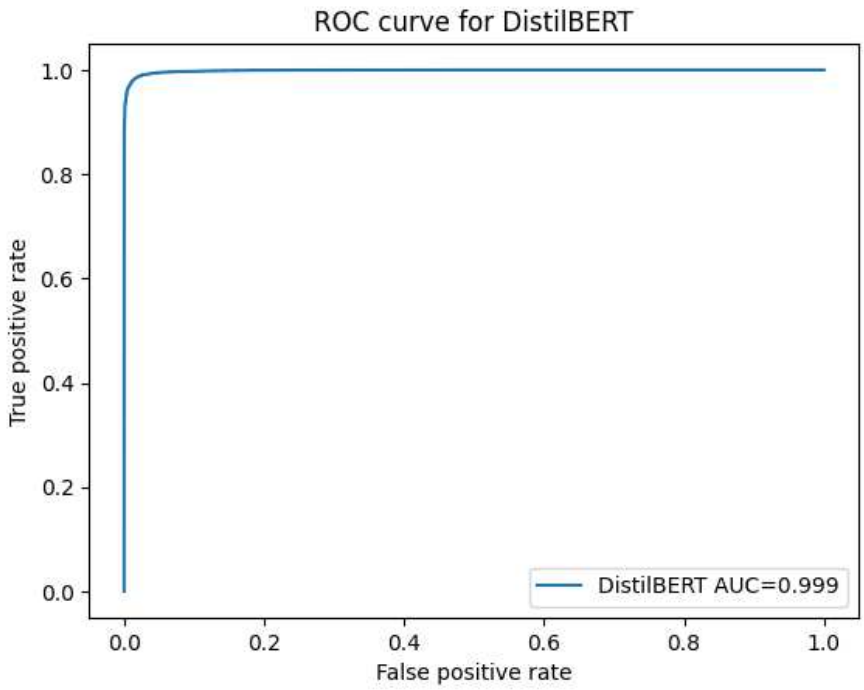
Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.93	0.96	15 000
1	0.93	1.00	0.96	15 000
prosjek	0.96	0.96	0.96	30 000

Tablica 6.10. Klasifikacijski izvještaj za klasifikator DistilBERT

6.3.1. Rezultati klasifikatora DistilBERT nad dodatnim podatcima za ispitivanje

Rad klasifikatora DistilBERT dodatno je vrednovan koristeći dva dodatna skupa podataka opisana u poglavlju 3.1.1. Prvi skup podataka većinski sadrži eseje koje su napisali učenici, odnosno primjere iz negativne klase. Matrica zabune klasifikatora DistilBERT za ovaj skup podataka prikazana je slikom 6.11. Model je ispravno klasificirao sve primjere pozitivne klase, a pogrešno klasificiranih primjera iz negativne klase ima 311.

Na osnovi elemenata matrice konfuzije (zabune) izračunate su metrike preciznosti, odziva i F1 mjera za svaku klasu. Rezultati ovih metrika prikazani su tablicom 6.11. Vrlo nisku vrijednost preciznosti za pozitivnu klasu i vrlo visoku vrijednost preciznosti za negativnu klasu uvjetovala nebalansiranost broja primjera iz različitih klasa u ovom skupu podataka. Realno stanje pokazuje metrika odziva koja iznosi 1 za primjere iz pozitivne



Slika 6.10. ROC krivulja za klasifikator DistilBERT

Klasa	**Preciznost**	**Odziv**	**F1 mjera**	Broj primjera
0	1.00	0.77	0.87	1375
1	0.01	1.00	0.02	3
težinski prosjek	1.00	0.77	0.87	1378

Tablica 6.11. Klasifikacijski izvještaj za klasifikator DistilBERT za skup podataka koji ispituje negativnu klasu

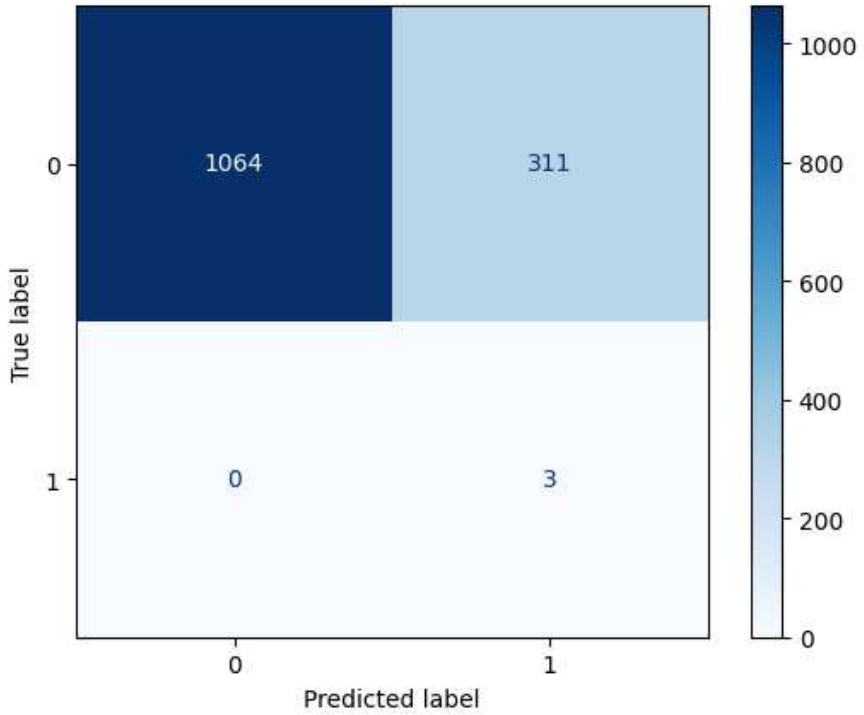
klase te 0.77 za primjere iz negativne klase.

Sljedeći skup podataka koristio se za ispitivanje rada klasifikatora na pozitivno označenim primjerima. Matrica zabune za ovih 3611 primjera prikazana je slikom 6.12. Klasifikator DistilBERT pogrešno je klasificirao 444 primjera.

Na osnovi vrijednosti iz matrice zabune izračunata je mjera odziva za sve pozitivne primjere, odnosno za cijeli skup podataka. Odziv iznosi 0.88, a rezultat se može vidjeti u tablici 6.12.

Klasa	**Odziv**	Broj primjera
1	0.88	3611

Tablica 6.12. Klasifikacijski izvještaj za klasifikator DistilBERT za skup podataka koji ispituje pozitivnu klasu

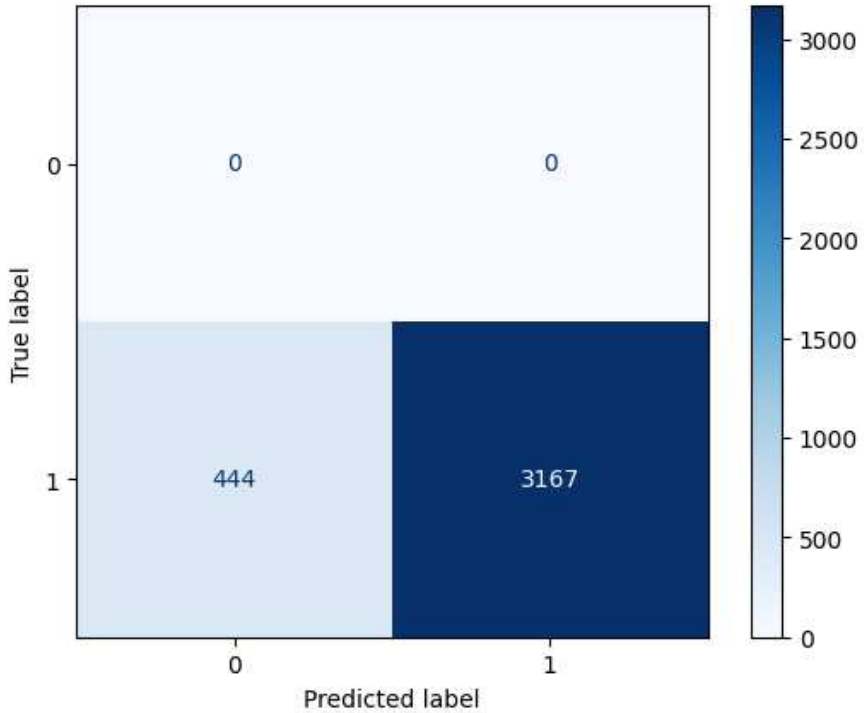


Slika 6.11. Matrica zabune klasifikatora DistilBERT za skup podataka koji ispituje negativnu klasu

6.4. Rasprava

Sva tri klasifikatora korištena za klasifikaciju teksta generiranog velikim jezičnim modelom pokazala su vrlo visoke rezultate svih metrika kojima su vrednovani. Ipak, postoje sitne razlike u rezultatima koje će se detaljnije raspraviti.

Općenito, za problem prepoznavanja teksta koje je generirao veliki jezični model bitnije je ispravno prepoznati pozitivne primjere, tj. one primjere koji su označeni kao da su generirani nekim velikim jezičnim modelom. Drugim riječima, manje je bitno kada je primjer označen kao lažno pozitivan (FP) nego kada je označen kao lažno negativan (FN). Razlog tome je što je za ovaj problem manje skupo vršiti dodatne provjere da se utvrdi da neki primjer zapravo nije generiran velikim jezičnim modelom, nego imati slučaj kada neki tekst neopaženo prođe kao da ga je napisao čovjek, a nije. U slučaju da je neki čovjek uistinu napisao tekst, a taj tekst je bio klasificiran kao da ga je generirao veliki jezični model, autor teksta može se žaliti na rezultate te dodatnim provjerama dokazati valjanost teksta. U slučaju da je tekst bio generiran, a model ga je klasificirao kao da ga je napisao čovjek, navodni autor teksta je uspješno prevario ispitivanje te je malo vjerojatno da će podnijeti žalbu na rezultate. Iz tog razloga pri vrednovanju klasifikatora



Slika 6.12. Matrica zabune klasifikatora DistilBERT za skup podataka koji ispituje pozitivnu klasu

bitno je obratiti posebnu pažnju na količinu lažno pozitivnih primjera u odnosu na lažno negativne i na mjeru odziva koja ispituje kvalitetu predikcija za pozitivnu klasu.

Na osnovi matrica zabuna za sve klasifikatore, klasifikator DistilBERT ima najlošije rezultate s 52 lažno negativna primjera te 1062 lažno pozitivna primjera. Klasifikator BERT ima najmanji broj lažno negativnih primjera, njih 31, ali ima najviše lažno pozitivnih primjera, njih 1298. Klasifikator RoBERTa ima nešto više lažno negativnih primjera od klasifikatora BERT, njih 48, ali ima znatno manje lažno pozitivnih primjera, njih 989.

Rezultati ostalih metrika za sva tri klasifikatora grupirani su i prikazani tablicom 6.13. Na osnovi ovih rezultata može se vidjeti da sva tri klasifikatora daju vrlo dobre rezultate na skupu za ispitivanje. Klasifikator BERT pokazuje najbolje rezultate za mjere odziva i površine ispod ROC krivulje što je bitno za problem prepoznavanja teksta generiranog velikim jezičnim modelom, ali klasifikator RoBERTa pokazuje najbolje rezultate za mjere točnosti i mjeru F1 koja je harmonijska sredina odziva i preciznosti što ukazuje na sveukupan rad modela, a ne samo za pozitivnu klasu.

Nadalje, grupirani rezultati mjere odziva klasifikatora za dodatne podatke za ispitiva-

	Točnost	Preciznost	Odziv	F1 mjera	AUC
BERT	0.956	0.920	0.998	0.957	0.9990
RoBERTa	0.965	0.938	0.997	0.966	0.9989
DistilBERT	0.963	0.934	0.997	0.964	0.9987

Tablica 6.13. Rezultati metrika za klasifikatore BERT, RoBERTa i DistilBERT

	Odziv za klasu 0	Odziv za klasu 1
BERT	0.91	0.82
RoBERTa	0.93	0.90
DistilBERT	0.77	0.88

Tablica 6.14. Rezultati mjere odziva za dodatne skupove za ispitivanje klasifikatora BERT, RoBERTa i DistilBERT

nje prikazani su u tablici 6.14. Klasifikator RoBERTa pokazuje najbolje rezultate za obje klase primjera. Iako klasifikator DistilBERT ima bolje rezultate od klasifikatora BERT na primjerima iz pozitivne klase, pokazuje znatno lošije rezultate za negativnu klasu. To znači da klasifikator BERT sveukupno pokazuje bolji rad na novim podatcima nego klasifikator DistilBERT. S druge strane, bitno je napomenuti da je proces učenja klasifikatora DistilBERT čak četverostruko brži nego proces učenja klasifikatora RoBERTa i BERT jer model DistilBERT ima znatno manji broj parametara. S time se pri procesu učenja troši znatno manja količina resursa. To pokazuje da u slučaju manje dostupnih resursa i vremena, klasifikator DistilBERT može biti dobra alternativa modelima s više parametara poput modela RoBERTa i BERT. U slučaju kada je dostupno više resursa i vremena, pokazalo se da je klasifikator RoBERTa optimalan izbor zbog generalno najboljih rezultata za problem prepoznavanja teksta generiranog velikim jezičnim modelom.

7. Zaključak

Područje obrade prirodnog jezika obuhvaća zadatke prevođenja teksta, odgovaranja na pitanja, klasifikacije teksta, generiranja teksta i slične. Za savladavanje ovakvih problema potrebne su velike količine podataka u obliku teksta. U svrhu rješavanja problema u području obrade prirodnog jezika, razvijeni su brojni veliki jezični modeli koji su naučeni nad golemim količinama podataka i koji se sastoje od velikog broja parametara. U posljednje vrijeme, ovakvi modeli porasli su na popularnosti zbog svoje visoke preciznosti i jednostavnosti korištenja. Ljudi koriste jezične modele koji su sposobni generirati tekst za svakodnevne životne probleme. Ovakva učestala uporaba velikih jezičnih modela dovela je i do korištenja istih u krive svrhe, predstavljajući tekst koji je generiran nekim velikim jezičnim modelom kao da je napisan od strane čovjeka. Prevare ovog karaktera postalo je kompleksno za uočiti ljudskom intervencijom pa se stvorila potreba za ostvarenjem sustava koji će imati mogućnost diferenciranja teksta generiranog velikim jezičnim modelima od teksta kojeg je zapravo napisao čovjek.

U ovom radu programski su ostvareni klasifikatori čiji je zadatak prepoznati tekst generiran velikim jezičnim modelom. Koristili su se modeli BERT, RoBERTa i DistilBERT koji su zasnovani na transformerima. Njihov rad ispitan je koristeći skup podataka za ispitivanje koji je izdvojen iz skupa podataka *„GPT Wiki Intro”* koji se koristio za učenje modela te na dodatnim podatcima za ispitivanje koji su generirani različitim velikim jezičnim modelima. Klasifikator s najvećom razinom mjere odziva za dodatan skup za ispitivanje koja iznosi 0.9 bio je klasifikator RoBERTa. Slijedi ga klasifikator DistilBERT s razinom odziva od 0.88, koji se također ističe sa znatno manjim brojem parametara trošeći manje resursa pri treniranju. Najlošije rezultate mjere odziva za dodatan skup za ispitivanje pokazao je klasifikator BERT s iznosom od 0.82.

Literatura

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, i I. Polosukhin, "Attention is all you need", http://arxiv.org/abs/1706.03762, 2017., [Mrežno; stranica posjećena: travanj 2024.].

[2] J. Devlin, M. Chang, K. Lee, i K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding", http://arxiv.org/abs/1810.04805, 2018., [Mrežno; stranica posjećena: travanj 2024.].

[3] R. Tang, Y.-N. Chuang, i X. Hu, "The science of detecting llm-generated texts", https://arxiv.org/pdf/2303.07205, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[4] R. Koike, M. Kaneko, i N. Okazaki, "Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples", https://arxiv.org/pdf/2307.11729v3, 2024., [Mrežno; stranica posjećena: travanj 2024.].

[5] T. Munyer, A. Tanvir, A. Das, i X. Zhong, "Deeptextmark: A deep learning-driven text watermarking approach for identifying large language model generated text", https://arxiv.org/pdf/2305.05773, 2024., [Mrežno; stranica posjećena: svibanj 2024.].

[6] A. Doherty, "Splitting data into 'train', 'validation' and 'test' sets", https://harksys.com/blog/splitting-data-into-train-validation-and-test-sets/, 2022., [Mrežno; stranica posjećena: lipanj 2024.].

[7] Aaditya Bhat, "Gpt-wiki-intro (revision 0e458f5)", https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro, 2023., [Mrežno; stranica posjećena: travanj 2024.]. https://doi.org/10.57967/hf/0326

[8] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, i N. Fiedel, "Palm: Scaling language modeling with pathways", https://arxiv.org/pdf/2204.02311, 2022., [Mrežno; stranica posjećena: svibanj 2024.].

[9] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, Étienne Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, i G. Penedo, "The falcon series of open language models", https://arxiv.org/pdf/2311.16867, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[10] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, i G. Lample, "Llama: Open and efficient foundation language models", https://arxiv.org/pdf/2302.13971, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[11] N. Bhoi, "Stemming vs lemmatization in nlp", https://nirajbhoi.medium.com/stemming-vs-lemmatization-in-nlp-efc280d4e845, 2022., [Mrežno; stranica posjećena: svibanj 2024.].

[12] HuggingFace, "How do transformers work?" https://huggingface.co/learn/nlp-course/en/chapter1/4, [Mrežno; stranica posjećena: lipanj 2024.].

[13] A. Saleem, "Transformer models: the future of natural language processing", https://datasciencedojo.com/blog/transformer-models/, 2023., [Mrežno; stranica posjećena: lipanj 2024.].

[14] C. Jeeva, “Pre-training the bert model”, https://www.scaler.com/topics/nlp/pre-training-bert/, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[15] ——, “Next sentence prediction with bert”, https://www.scaler.com/topics/nlp/bert-next-sentence-prediction/, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[16] A. R, “How does bert nlp optimization model work?” https://www.turing.com/kb/how-bert-nlp-optimization-model-works, [Mrežno; stranica posjećena: svibanj 2024.].

[17] D. P. Kingma i J. Ba, “Adam: A method for stochastic optimization”, https://arxiv.org/pdf/1412.6980, 2017., [Mrežno; stranica posjećena: svibanj 2024.].

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, i V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach”, https://arxiv.org/pdf/1907.11692, 2019., [Mrežno; stranica posjećena: travanj 2024.].

[19] V. Efimov, “Large language models: Roberta — a robustly optimized bert approach”, https://towardsdatascience.com/roberta-1ef07226c8d8, 2023., [Mrežno; stranica posjećena: svibanj 2024.].

[20] K. Kacprzak, “Roberta vs. bert: Exploring the evolution of transformer models”, https://dsstream.com/roberta-vs-bert-exploring-the-evolution-of-transformer-models/, 2024., [Mrežno; stranica posjećena: svibanj 2024.].

[21] V. Sanh, L. Debut, J. Chaumond, i T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter”, https://arxiv.org/pdf/1910.01108, 2020., [Mrežno; stranica posjećena: lipanj 2024.].

[22] C. Bucila, R. Caruana, i A. Niculescu-Mizil, “Model compression”, sv. 2006, 08 2006., str. 535–541. https://doi.org/10.1145/1150402.1150464

[23] G. Hinton, O. Vinyals, i J. Dean, “Distilling the knowledge in a neural network”, https://arxiv.org/pdf/1503.02531, 2015., [Mrežno; stranica posjećena: lipanj 2024.].

[24] S. Narkhede, “Understanding confusion matrix”, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 2018., [Mrežno; stranica posje-

ćena: svibanj 2024.].

[25] D. L. Streiner i J. Cairney, "What's under the roc? an introduction to receiver operating characteristics curves." *Canadian journal of psychiatry*, sv. 52, br. 2, str. 121–8, veljača 2007. https://doi.org/10.1177/070674370705200210

[26] J. Herlocker, J. Konstan, L. Terveen, J. C. Lui, i T. Riedl, "Evaluating collaborative filtering recommender systems", *ACM Transactions on Information Systems*, sv. 22, str. 5–53, 01 2004. https://doi.org/10.1145/963770.963772

[27] "Auc roc curve in machine learning", https://www.geeksforgeeks.org/auc-roc-curve/, 2024., [Mrežno; stranica posjećena: lipanj 2024.].

Sažetak

Prepoznavanje teksta generiranog velikim jezičnim modelom

Maria Fain

Veliki jezični modeli koriste se za probleme iz područja obrade prirodnog teksta poput klasifikacije, generiranja, sažimanja teksta i slično. Zlouporabom generativnih velikih jezičnih modela stvorila se potreba za sustavima sposobnima prepoznati tekst generiran velikim jezičnim modelom među tekstovima koje su napisali ljudi. Za rješavanje ovog problema programski su ostvareni klasifikatori BERT, RoBERTa i DistilBERT naučeni koristeći skup podataka *„GPT Wiki Intro”*. Dio tekstova u skupu za učenje generiran je velikim jezičnim modelom *Curie*. Nakon što su klasifikatori istrenirani, provedeno je vrednovanje uspješnosti njihovih predikcija nad skupom podataka za ispitivanje koristeći razne metrike. Klasifikatori su dodatno vrednovani koristeći tekstove koje su generirali različiti veliki jezični modeli.

Ključne riječi: obrada prirodnog jezika; veliki jezični modeli; klasifikacija teksta; BERT; RoBERTa; DistilBERT; modeli zasnovani na transformerima; matrica zabune

Abstract

Detection of Text Generated by a Large Language Model

Maria Fain

Large language models (LLMs) are used for problems in the field of natural language processing such as text classification, generating or summarizing text and others. The misuse of generative LLMs created a need for systems which would be capable of detecting texts generated by an LLM among texts written by humans. For solving this problem, classifiers BERT, RoBERTa and DistilBERT were implemented and trained using the *"GPT Wiki Intro"* dataset. A share of the texts in the training dataset was generated by the *Curie* LLM. After the classifiers were trained, their predictions on the testing dataset were validated using various metrics. The classifiers were additionally validated using texts that were generated by different LLMs.

Keywords: natural language processing; large language models; text classification; BERT; RoBERTa; DistilBERT; transformer models; confusion matrix