UNIVERZITET U BANJOJ LUCI

ELEKTROTEHNIČKI FAKULTET

PRIMJENA MAŠINSKOG UČENJA ZA OPISIVANJE SLIKA POMOĆU TEKSTA

MASTER RAD

Mentor:
Prof. dr Zoran Đurić

Kandidat:
Aleksije Mićić

Banja Luka, maj 2025.

UNIVERSITY OF BANJA LUKA

FACULTY OF ELECTRICAL ENGINEERING

THE APPLICATION OF MACHINE LEARNING FOR IMAGE CAPTIONING

MASTER THESIS

Mentor:
Prof. Zoran Đurić, PhD

Candidate:
Aleksije Mićić

Banja Luka, May 2025.

Mentor: Dr Zoran Đurić, redovni profesor,
Elektrotehnički fakultet,
Univerzitet u Banjoj Luci

Naziv master rada: Primjena mašinskog učenja za opisivanje slika pomoću teksta

Rezime: Moderni informacioni sistemi za e-trgovinu integrišu sisteme za preporuku proizvoda kako bi upotpunili korisničko iskustvo. U ovom radu su analizirane primjene opisivanja slika pomoću tehnika mašinskog učenja, s ciljem generisanja boljih preporuka za e-trgovine koje se bave prodajom odjevnih predmeta. Posebna pažnja je posvećena problemima sa tradicionalnim pristupom za pretragu proizvoda, koja je obično bazirana na analizi sličnosti slika. U praktičnom dijelu rada evaluirani su opisi generisani pomoću osam različitih modela, pri čemu su se najbolje pokazali opisi generisani sa velikim jezičkim modelima. Nakon toga je evaluirano predloženo rješenje koje kombinuje tradicionalni pristup sa analizom sličnosti opisa na prethodno opisanim problemima, gdje se pokazala opravdanost korištenja novog pristupa. Na kraju rada dati su mogući pravci daljnjeg istraživanja.

Ključne riječi: Sistemi za preporuku, opisivanje slika pomoću teksta, pretraga unutar prodavnice, pretraga od potrošača do prodavnice, veliki jezički modeli

Naučna oblast: Inženjerstvo i tehnologija

Naučno polje: Elektrotehnika, elektronika i informaciono inženjerstvo

Klasifikaciona oznaka: T 120

Mentor: Zoran Đurić, PhD, full professor,
Faculty of Electrical Engineering,
University of Banja Luka

Master thesis title: The application of machine learning for image captioning

Abstract: Modern information systems for e-commerce integrate product recommendation systems to enhance the user experience. This thesis analyzes the application of image captioning using machine learning to generate better recommendations for e-commerce platforms that sell clothing. Special attention is given to issues with traditional product retrieval methods based on image similarity analysis. Image captions generated by eight different models were evaluated, with large language models performing the best. Following this, the proposed solution, which combines the traditional retrieval approach with similarity analysis of captions, was evaluated on the previously described problems, demonstrating the validity of using the new approach. Finally, potential directions for further research are provided.

Keywords: Recommendation systems, image captioning, shop-to-shop retrieval, consumer-to-shop retrieval, large language models

Scientific field: Engineering and technology

Scientific area: Electrical engineering, electronics and information engineering

Classification label: T 120

Spisak korištenih skraćenica

AI	Artificial Intelligence
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
DF2	DeepFashion2
DL	Deep Learning
FFNN	Feedforward Neural Network
FP	False Positive
FPR	False Positive Rate
GRU	Gated Recurrent Unit
LCS	Longest Common Subsequence
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Modeling
NAS	Neural Architecture Search
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OvO	One-Versus-One
OvR	One-Versus-Rest
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

Lista slika

Lista tabela

Sadržaj

1. UVOD

Ubrzanim razvojem informacionih sistema i računarskih mreža došlo je do ekspanzije broja korisnika računara širom svijeta. Samim tim, višestruko se povećala i količina multimedijalnog sadržaja koja se generiše i dijeli ovim putem. Svakodnevno se generišu ogromne količine podataka u različitim oblicima, pri čemu slike čine značajan dio digitalnog ekosistema. Međutim, mnoge slike ne sadrže dovoljno pratećih informacija, tzv. meta-podataka (eng. *metadata*), koje bi omogućile njihovu preciznu klasifikaciju, pretragu ili kontekstualnu analizu, a što otežava njihovu efikasnu upotrebu u različitim aplikacijama.

Paralelno s tim, mnoge aktivnosti koje su se nekada obavljale isključivo u fizičkom prostoru sada su dobile svoje digitalne ekvivalente. Od društvenih interakcija do poslovanja, sve više aspekata svakodnevnog života premješta se u virtuelno okruženje, pri čemu je elektronska trgovina (e-trgovina) jedan od najznačajnijih segmenata. Kupovina putem interneta postala je ne samo praktična, već i nezaobilazna u modernom društvu, omogućavajući korisnicima pristup širokom spektru proizvoda iz različitih kategorija. Ovakav vid kupovine omogućava generisanje preporuka za proizvode na osnovu korisničkih akcija, što može pozitivno uticati na zaradu prodavača [1-3]. S druge strane, korisnička očekivanja, vezana za kvalitet preporuka, su sa godinama samo porasla, a od kvaliteta preporuka uveliko zavisi cjelokupno korisničko iskustvo, naročito u e-trgovinama koje se bave prodajom odjevnih predmeta. Za pronalazak sličnih i povezanih proizvoda, posebno za pretrage unutar prodavnice (eng. *shop-to-shop retrieval*) i od potrošača do prodavnice (eng. *consumer-to-shop retrieval*), u daljnjem tekstu pretraga prodavnice, koristi se analiza sličnosti slika [4-5]. Kao upit uzima se slika na kojoj su prikazani odjevni predmeti od interesa, a kao rezultat pretrage dobijaju se dostupni slični i povezani odjevni predmeti iz kataloga proizvoda e-trgovine. Ovaj pristup veoma brzo generiše preporuke koje su, u zavisnosti od složenosti upita, vizuelno slične predmetima koji su dio upita. Informacije koje su sadržane u upitu je potrebno pretvoriti u odgovarajući numerički format koji je razumljiv računaru, za te svrhe se koriste modeli mašinskog učenja specijalizovani za analizu sličnosti slika. Pomoću njih se za upit dobija odgovarajuća vektorska reprezentacija. Memorijski zahtjevi za čuvanje vektorske reprezentacije slika svih proizvoda su relativno mali, čak kada je riječ i o velikim katalozima sa milionima proizvoda. Ipak, ovakav pristup ima i svoje mane. Proces prikupljanja i labelisanja podataka koji će se koristiti tokom obučavanja modela je mukotrpan i iziskuje mnogo vremena, ali i zahtjeva obučene anotatore, zbog čega može biti skup. Takođe, nije dovoljno samo obučiti model koji vrši analizu sličnosti slika odjevnih predmeta, jer dati model kao ulaz podrazumijevano dobija pojedinačne predmete. Zbog toga je potrebno obučiti i model koji detektuje pojedinačne predmete na slici, tako što određuje njihov granični okvir (eng. *bounding box*) i dodijeli im odgovarajuću klasu [4-6]. U zavisnosti od složenosti upita, može doći do detekcije lažno pozitivnih predmeta, npr. kada je upit slika majice koja ima grafički print na kom su prikazani drugi ljudi koji na sebi imaju odjevne predmete za koje se onda takođe traže slični proizvodi. Takođe, ako je upit složeniji, odnosno ako je na slici osoba koja nosi višeslojnu garderobu, tada je često nemoguće identifikovati sve odjevne predmete. Kada je detekcija predmeta uspješna, problem ponekad predstavlja i to što model pronalazi vizuelno sličnu odjeću, bez razumijevanja konteksta. Ako je npr. upit košulja koju nosi neka osoba, onda

će košulje iz kataloga koje su na slici savijene imati manju sličnost, čak i kada je riječ o identičnoj košulji. Ovaj problem je još više izražen, kada je kao upit data majica sa nekim likom iz popularne kulture, npr. lik iz filma, jer će se kao rezultat dobiti majice koje prikazuju druge, vizuelno slične likove, ne nužno povezane sa datim likom, gdje bi uobičajeno bilo adekvatnije prikazati majice sa drugim likovima povezanim sa datim likom, ili čak majice sa natpisima vezani za istu franšizu. Za prevazilaženje ovih mana je u ovom radu predložen pristup koji kombinuju analizu sličnosti slika sa opisima slika generisanim pomoću tehnika mašinskog učenja.

Ovaj rad se bavi primjenom opisivanja slika pomoću tehnika mašinskog učenja za poboljšanje korisničkih preporuka, posebno za problem pretraživanja prodavnice. Jedan od ciljeva ovog istraživanja je da se obrade, opišu i objasne tehnike mašinskog učenja koje se koriste za opisivanje slika pomoću teksta, kao i metrike koje se koriste za evaluaciju ovih sistema. Glavni cilj rada bio je razvoj sistema za preporuke u okviru problema pretrage prodavnica, koji kombinuje analizu sličnosti slika i tekstualne opise kako bi korisnicima pružio što preciznije preporuke u skladu sa njihovim očekivanjima.

U drugom poglavlju su opisane teorijske osnove iz oblasti mašinskog učenja, čije razumijevanje je neophodno za rješavanje bilo kog složenijeg problema. Date su definicije osnovnih pojmova, kao i kratak osvrt na postupak obučavanja modela klasifikatora. U ovom poglavlju dat je opis osnovnih metrika za evaluaciju, te opis tehnika klasterizacije i dotreniranje.

U trećem poglavlju su opisane duboke neuronske mreže (eng. *deep neural networks*), koje se koriste za rješavanje brojnih problema iz oblasti mašinskog učenja, uključujući analizu sličnosti slika i detekciju objekata. Neuronske mreže čine osnovu arhitekture mnogih savremenih modela za vektorsku reprezentaciju teksta, a neke od tih reprezentacija su obrađene u petom poglavlju. Prvo su opisane jednostavne mreže bez povratnih veza, potom je dat opis složenijih konvolucionih neuronskih mreža (eng. *convolutional neural networks* - CNN) i rekurentnih neuronskih mreža (eng. *recurrent neural network* - RNN) koje se u sprezi mogu koristiti za generisanje opisa slika.

U četvrtom poglavlju obrađena je arhitektura transformatora, koja predstavlja unapređenje u odnosu na prethodno opisane neuronske mreže. Objašnjen je osnovni princip njihovog rada, nakon toga je prikazana arhitektura i način funkcionisanja modela koji se koriste za opisivanje slika pomoću teksta. Na kraju poglavlja su opisani veliki jezički modeli, sa posebnim fokusom na multimodalne velike jezičke modele, koji omogućavaju generisanje jako detaljnih i kvalitetnih opisa, koji se onda mogu iskoristiti za različite zadatke, kao što je pretraga prodavnice.

Peto poglavlje se bavi obradom prirodnog jezika. Prvo su opisane različite vrste vektorskih reprezentacija teksta, počevši od najprostijih baziranih na frekvenciji pojavljivanja riječi, do onih složenijih baziranih na ugrađivanju riječi i kontekstualnim jezičkim modelima. Nakon toga su opisane standardne metrike koje se u literaturi koriste za evaluaciju kvaliteta generisanih opisa slika, pri čemu je dat osvrt na njihove vrline i mane.

U šestom poglavlju opisan je problem pretrage prodavnice proizvoda. Prvo je opisana i objašnjena arhitektura tradicionalnog sistema za pretragu prodavnice proizvoda, koja se bazira na detekciji objekata i analizi sličnosti slika. Navedeni su i ilustrovani najveći nedostaci tradicionalnog pristupa, gdje se razlikuju nedostaci koji potiču od modela za detekciju objekata i nedostaci koji potiču od modela za analizu sličnosti slika. Potom je dat opis predloženog rješenja koje kombinuje tradicionalni pristup baziran na analizi sličnosti slika sa opisima slika generisanim pomoću tehnika mašinskog učenja.

U sedmom poglavlju su opisani eksperimentalni rezultati rada. Objašnjen je postupak obučavanja modela za tradicionalni pristup pretrazi proizvoda, dat je i opis korištenih modela i skupova podataka za ovaj dio sistema. Potom su opisani skupovi podataka, kao i modeli koji su korišteni za problem generisanja opisa slika odjevnih predmeta. Takođe, opisan je i proces *prompt engineering*-a koji se koristi kod velikih jezičkih modela u cilju dobijanja što kvalitetnijih odgovora. Zatim su dati rezultati evaluacije kvaliteta opisa na metrikama koje su prethodno opisane u petom poglavlju. Ovdje su se najbolje pokazali veliki jezički modeli, što je bilo i očekivano. Konačno, dati su rezultati evaluacije primjene kombinovanog pristupa na problemima kod pretrage koji su prethodno opisani u šestom poglavlju rada.

Na kraju rada data su zaključna razmatranja i prijedlozi mogućih unapređenja, kao i pravci budućeg istraživanja.

2. MAŠINSKO UČENJE

Vještačka inteligencija (eng. *artificial intelligence* - AI) je oblast računarstva koja se bavi stvaranjem sistema koji imitiraju ljudsku inteligenciju. AI obuhvata više oblasti kao što su ekspertski sistemi, obrada prirodnog jezika, mašinsko učenje itd [7].

Mašinsko učenje (eng. *machine learning* - ML) je grana AI koja se bavi razvojem algoritama i modela koji omogućavaju računarima da uče iz podataka i poboljšavaju svoje performanse na osnovu stečenog iskustva. Tako računari uče da obavljaju zadatke za koje nisu bili eksplicitno programirani. ML se koristi za rješavanje problema koji se ne mogu opisati jednostavnim skupom pravila, ali su ljudima intuitivni, kao što su prepoznavanje objekata, razumijevanje govora, opisivanje slika pomoću teksta itd [8].

Duboko učenje (eng. *deep learning* - DL) je grana ML koja koristi višeslojne (vještačke) neuronske mreže [9]. O DL će biti više riječi u narednim poglavljima.

Među najčešćim zadacima koji se rješavaju primjenom mašinskog učenja izdvajaju se klasifikacija i klasterizacija, koji nalaze široku primjenu u različitim oblastima, uključujući i opisivanje slika pomoću teksta i opisani su u nastavku.

2.1. Klasifikacija

Klasifikacija je postupak gdje se na osnovu atributa (obilježja) ulaznog uzorka odredi klasa kojoj uzorak pripada. Skup klasa je predefinisan i sadrži konačno mnogo elemenata $\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_\mathrm{K}\}$. Kada skup klasa sadrži samo dva elementa tada je riječ o binarnoj klasifikaciji, a kada je $K > 2$, radi se o višeklasnoj klasifikaciji (klasifikaciji u više klasa) [10].

Klasifikacija se može vršiti ručno, korišćenjem nekog skupa pravila i obučavanjem modela klasifikatora pomoću ML. U kontekstu ML klasifikator predstavlja neki algoritam, a model klasifikatora je konkretna instanca tog klasifikatora obučena na nekim podacima.

Binarni klasifikator uzorak svrstava u jednu od dvije klase, pri čemu se kao rezultat klasifikacije vraća klasa u koju je dati uzorak svrstan i vjerovatnoća da uzorak pripada datoj klasi. Za klasifikaciju u više klasa može se koristiti i binarna klasifikacija pomoću tehnika jedan-protiv-svih (eng. *one-versus-rest* - OvR) i jedan-protiv-jedan (eng. *one-versus-one* - OvO). OvR je tehnika gdje se obuči K binarnih klasifikatora, za svaku od K klasa, pri čemu dati klasifikatori vraćaju vjerovatnoću da li uzorak pripada jednoj od datih klasa ili ne, a konačna klasa se određuje na osnovu toga koja od klasa je imala najveću vjerovatnoću. Prednost OvR u odnosu na običnu klasifikaciju u više klasa je što je konceptualno jednostavna za razumijevanje, kao i u brzini klasifikacije, posebno ako se klasifikacija može izvršiti paralelno, a nedostatak je što je potrebno obučiti više klasifikatora. OvO je tehnika gdje je potrebno obučiti još više klasifikatora nego kod OvR, odnosno obučava se klasifikator za svaki par klasa. Klasifikatori vraćaju vjerovatnoću da uzorak pripada jednoj od dviju klasa, a konačna klasa se odredi tako što se uzima ona klasa u koju se uzorak najčešće svrstava. Prednost OvO je u intuitivnosti, a mana je u još većem broju klasifikatora [11].

Ako su klase međusobno disjunktne tada se uzorak svrstava u tačno jednu klasu, a kada klase nisu međusobno disjunktne tada se radi o problemu obilježavanja (eng. *tagging*) i svakom uzorku se može dodijeliti više klasa [10].

Formalno, model klasifikatora je funkcija $f: X \to C$, koja dati uzorak $x \in X$ svrstava u klasu $\hat{y} = f(x) \in C$. Funkcija f se naziva funkcija odlučivanja i ona određuje način na koji se neki uzorak klasifikuje u određenu klasu. Modeli obično imaju parametre od kojih neki utiču na strukturu, obučavanje i rad klasifikatora i nazivaju se hiperparametrima, koje je potrebno odrediti prije obučavanja. Ostali parametri se određuju tokom postupka obučavanja [10].

Suštinski cilj obučavanja modela klasifikatora jeste da se odrede pogodne vrijednosti za parametre klasifikatora. Postoje tri glavne vrste obučavanja [9, 11]:

- Nadgledano obučavanje (eng. *supervised learning*), koristi podatke za koje je unaprijed poznata pripadnost klasama. Koristi se za obučavanje modela klasifikatora, ali i za obučavanje modela regresije. Modeli regresije se koriste za predviđanje numeričkih vrijednosti kao što su berzanska cijena nafte, vrijeme koje će korisnik provesti na nekoj internet stranici i sl. Postoje razni modeli regresije kao što su linearna regresija (eng. *linear regression*), stabla odlučivanja (eng. *decision trees*), šuma slučajnih stabala (eng. *random forest*) itd.
- Nenadgledano obučavanje (eng. *unsupervised learning*), koristi podatke za koje nije poznata pripadnost klasama. Često se koristi za klasterizaciju podataka, redukciju dimenzionalnosti i generisanje novih podataka na osnovu starih.
- Podržano obučavanje (eng. *reinforcement learning* - RL), koristi se kada je potrebno obučiti računar da uradi neke akcije za koje ni ljudi često ne znaju najbolji način na koji bi se izvele. RL se oslanja na povratne informacije u vidu kazni i nagrada.

Nadgledano obučavanje, dakle, zahtjeva postojanje klasifikovanog (labelisanog) skupa podataka. Ipak kreiranje takvog skupa podataka je dugoročno znatno brži postupak nego ručna klasifikacija ili klasifikacija na osnovu pravila [10].

2.1.1. Priprema podataka

Kvalitet modela klasifikatora zavisi od kvaliteta podataka koji su korišteni za obučavanje. Prije samog obučavanja potrebno je prikupiti i pripremiti podatke na adekvatan način. U oblasti ML-a se razlikuju dva osnovna tipa podataka [9]:

- Kategorički podaci, odnosno podaci koji nemaju brojnu vrijednost.
    - Nominalni podaci, odnosno podaci koji nemaju unaprijed određen redoslijed ili hijerarhiju. Primjeri uključuju boje, žanrove filmova, nazive gradova itd.
    - Ordinalni podaci, odnosno podaci koji imaju smislen redoslijed po kome se mogu sortirati. Primjeri uključuju vladanje učenika, stepen obrazovanja, vojni čin itd.
- Numerički podaci - podaci koji su predstavljeni numeričkim vrijednostima.

- Podaci o razmjeri - karakteriše ih prisustvo apsolutne nule, što znači da postoji tačka u kojoj određena veličina potpuno prestaje da postoji u fizičkom smislu. Primjeri takvih podataka su visina, težina i slične mjere.
- Intervalski podaci, to su podaci koji nemaju apsolutnu nulu. Primjer je kalendarski datum.

ML algoritmi operišu brojevima kao ulaznim podacima, stoga je neophodno transformisati sve kategoričke podatke, poput tekstualnih informacija, u numeričke. Za ovu konverziju se primjenjuju različite tehnike, kao što je *one-hot-encoding*. Ova tehnika predstavlja kategoričke podatke kao vektore nula i jedinica. Npr., u slučaju kategorija boja (crvena, zelena i plava), crvena boja se može označiti vektorom $[1,0,0]$, zelena vektorom $[0,1,0]$, dok plavu boju predstavlja vektor $[0,0,1]$. Prednost ove tehnike leži u jednostavnom prelasku iz kategoričkih u numeričke podatke. Međutim, postoje određene mane, uključujući povećanje dimenzionalnosti, posebno kada postoji veliki broj kategorija u skupu podataka, kao i gubitak informacija o redoslijedu i sličnostima između kategorija zbog ortogonalnosti *one-hot* vektora [11]. Naredna poglavlja će se baviti naprednijim tehnikama u ovom kontekstu.

Često se vrši normalizacija numeričkih podataka na određeni opseg, npr., interval $[0, 1]$ ili $[-1, 1]$. Ova praksa se primjenjuje kako bi se izbjegla situacija u kojoj određeni parametri imaju neujednačen uticaj na model tokom faze treniranja. Kada je riječ o obradi slika, često se koristi globalna normalizacija. Ova tehnika uključuje dijeljenje vrijednosti svakog piksela sa 255, čime se skalira vrijednost piksela na opseg između 0 i 1. Normalizacija se primjenjuje nezavisno na svaki od kanala. [12].

U cilju optimizacije procesa obuke klasifikatora, neophodno je eliminisati nepotrebna obilježja i one koji imaju minimalan uticaj na krajnje rezultate. Ovaj proces se naziva selekcija obilježja. Dodatno, potrebno je i eliminisati međusobno korelisane parametre primjenom tehnika za redukciju dimenzionalnosti, kao što je analiza glavnih komponenti (eng. *principle component analysis* - PCA).

2.1.2. Obučavanje modela klasifikatora

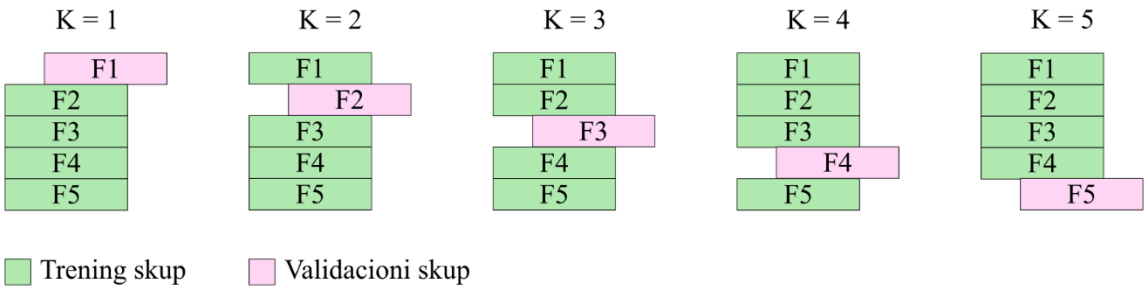
Model klasifikatora se obučava na skupu podataka koji se naziva trening skup. Inicijalno su vrijednosti parametara klasifikatora postavljene na početne vrijednosti (koje mogu biti i nasumično odabrane). Tokom obučavanja, klasifikator određuje pripadnost klasama za svaki od uzoraka pojedinačno. U slučaju da je uzorak pravilno klasifikovan, prelazi se na naredni, u suprotnom, ažurira se vrijednosti parametara modela klasifikatora tako da se smanji izračunata greška predikcije kako bi se u narednoj iteraciji dobilo bolje predviđanje [11]. Postupak optimizacije, odnosno smanjivanja greške predikcije detaljno je objašnjen na primjeru neuronskih mreža u narednim poglavljima.

Nakon što je model klasifikatora obučen na trening skupu, potrebno ga je evaluirati i u te svrhe se koristi testni skup podataka. Testni skup bi trebalo da sadrži uzorke kakve bi klasifikator susreo tokom realne upotrebe u produkciji, čime se garantuje da klasifikator može dobro da generalizuje. Potrebno je voditi računa da su testni i trening skup međusobno disjunktni, jer će u suprotnom klasifikator naučiti da pravilno klasifikuje testne uzorke, pa onda testni skup više neće biti relevantan za testiranje performansi modela klasifikatora [11].

Trening i testni skup se formiraju iz inicijalnog skupa podataka, koji je pribavljen za svrhe obučavanja modela klasifikatora. Uobičajeno je da se početni skup podijeli na trening i testni skup u odnosu $80:20$, ovakva podjela je bazirana na Paretovom pravilu[1], ali su još uobičajene razmjere $75:25$, $70:30$ i $60:40$. Ipak naučna istraživanja su pokazala da idealna razmjera zavisi od broja parametara modela p i data je sa odnosom $\sqrt{p}:1$, jer je potrebno mnogo više trening podataka da bi se estimirale vrijednosti parametara modela koji ima mnogo parametara, nego kod modela koji ima mali broj parametara [13].

Hiperparametri igraju ključnu ulogu u efikasnosti klasifikatora, a njihove optimalne vrijednosti se često pronalaze putem obučavanja više klasifikatora sa različitim konfiguracijama hiperparametara. Važno je napomenuti da prilagođavanje hiperparametara na osnovu rezultata na testnom skupu može dovesti do pristrasnih procjena sposobnosti modela za generalizaciju. Kako bi se izbjegla ova pristrasnost, uvodi se treći skup podataka, validacioni skup, koji je nezavisan od trening i test skupova. Validacioni skup omogućava objektivno procjenjivanje odabranih hiperparametara [10-11]. Uobičajeno je da se početni skup podataka podijeli na trening, test i validacioni skup u razmjeri $60:20:20$, ali je istraživanjima dokazno da je optimalna razmjera $p:\sqrt{p}:(\sqrt{p}+1)$. Ako je riječ o modelu sa $p=16$ parametara, tada se početni skup dijeli u razmjeri $16:4:5$ [13].

Ponekad je na raspolaganju samo ograničen skup podataka, pri čemu nije moguće doći do novih uzoraka. Tada je nezahvalno dijeliti početni skup na trening i testni, jer bi trening skup bio suviše mali da bi se mogao obučiti dobar klasifikator, a još manji bi bio testni skup, pa se na njemu ne bi mogli evaluirati rezultati obučavanja [8, 11]. U takvim situacijama se koriste tehnike unakrsne validacije (eng. *cross-validation*). Jedna takva tehnika je *K-fold* unakrsna validacija koja uključuje podjelu skupa podataka na K podskupova, nazvanih *fold*-ovi. Zatim se model trenira K puta, svaki put koristeći $K-1$ *fold*-ova za obuku i preostali *fold* za validaciju. Tokom svake od K iteracija koristi se drugi *fold* kao testni skup. Ovaj proces omogućava modelu da bude treniran i validiran na različitim podskupovima podataka. Na kraju, kada se završi obučavanje, uzima se srednja greška estimacije, svake od K iteracija. Ovaj proces je ilustrovan na slici 2.1. Mane ovakvog pristup su duži postupak obučavanje i što je dobijena greška samo estimacija prave testne greške.

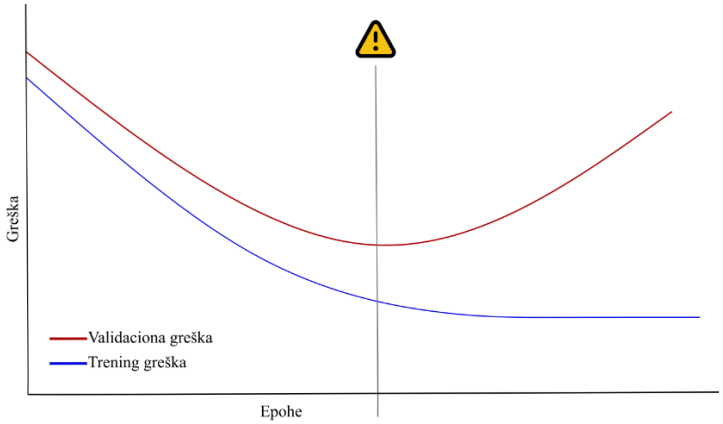


Slika 2.1: Obučavanje klasifikatora korišćenjem K-fold unakrsne validacije

[1] https://betterexplained.com/articles/understanding-the-pareto-principle-the-8020-rule/

Čest problem kada se model loše ponaša na testnom skupu, odnosno kada ne uspijeva da dobro generalizuje, može biti uzrokovan preprilagođenošću trening podacima (eng. *overfitting*) ili nedovoljnom prilagođenošću trening podacima (eng. *underfitting*). Nedovoljna prilagođenost trening podacima se lakše primjećuje i stoga je jednostavnija od dva problema za rješavanje; jednostavno je potrebno proširiti trening skup. S druge strane, preprilagođenost se može prevazići zaustavljanjem obučavanja prije nego što klasifikator počne učiti detalje koji su specifični za trening skup, ova tehnika se često naziva „ranim zaustavljanjem“, ili korišćenjem tehnika regularizacije [11].

Rano zaustavljanje se sprovodi tako što se trening greška poredi sa validacionom greškom, a to je greška koja se dobija evaluirajući performanse klasifikatora na validacionom skupu. Validaciona greška je ujedno i estimacija greške generalizacije, koja bi se dobila na testnom skupu odnosno u radu sa uzorcima iz realnog svijeta. Nakon svake epohe porede se trening i validaciona greška, u početku će obje greške opadati, sve do trenutka kada validaciona greška kreće ponovo da raste, što je znak da je model počeo da se preprilagođava trening skupu i da je potrebno zaustaviti obučavanje [14]. Odnos validacione i trening greške tokom obučavanja je ilustrovan na slici 2.2.



Slika 2.2 Grafički prikaz validacione i trening greške tokom epoha u procesu obučavanja klasifikatora

Korišćenjem ranog zaustavljanja prekida se postupak obučavanja prije nego što dođe do pojave preprilagođenosti trening skupu, ali je poželjno da obučavanje traje što duže kako bi se izvuklo što više informacija iz trening skupa. Zbog toga se koriste tehnike regularizacije kako bi što kasnije došlo do pojave preprilagođenosti tokom obučavanja, jer je tada moguće obučiti još bolji model [11].

Tehnike regularizacije spriječavaju da bilo koji od parametara modela postane suviše dominantan u odnosu na ostale, što je naročito značajno kako bi se umanjio uticaj parametara koji su manje bitni za krajnju klasifikaciju. Izuzimanje (eng. *dropout*) je popularna i efikasna tehnika za borbu protiv preprilagođavanja u neuronskim mrežama. Osnovna ideja je slučajno isključivanje čvorova i veza iz neuronske mreže tokom obučavanja. Postupak uključuje slučajno isključivanje skrivenih neurona sa vjerovatnoćom p, obučavanje pojednostavljene mreže, vraćanje uklonjenih neurona, ponovno slučajno isključivanje neurona i ponavljanje ovog procesa sve dok se ne dobiju optimalni parametri. Izuzimanje aproksimira efekat kombinovanja predviđanja različitih pojednostavljenih mreža, čime se djelimično sprječava

preprilagođavanje, uz značajno smanjenje računarskih resursa [11]. Neke druge često korišćene tehnike regularizacije uključuju: *weight decay*, *learning-rate decay*, *batch normalization*, itd.

2.1.3. Evaluacija modela klasifikatora

Postoji mnogo različitih mjera za evaluaciju modela klasifikatora. Osnovna mjera je tačnost (eng. *accuracy*). Tačnost se definiše kao omjer broja tačno klasifikovanih uzoraka prema ukupnom broju uzoraka u testnom skupu [10].

$$Tačnost = \frac{Broj\ tačno\ klasifikovanih\ uzoraka}{Ukupan\ broj\ uzoraka}. \tag{2.1}$$

Iako tačnost pruža osnovni uvid u performanse klasifikatora, ona ima značajan nedostatak. U problemima sa izraženim disbalansom klasa, njena pouzdanost može biti varljiva. Na primjer, kod detekcije bolesti na medicinskim slikama, od 100 slika samo jedna prikazuje bolest, dok ostalih 99 ne. Klasifikator koji sve slike klasifikuje kao zdrave ima tačnost od 99%. Međutim, ovaj klasifikator nije u stanju da detektuje bolest, što ga čini beskorisnim za stvarnu primjenu.

Da bi se dobio bolji uvid u performanse klasifikatora, neophodno je analizirati ne samo tačno klasifikovane uzorke, već i greške koje klasifikator pravi. Kada je u pitanju binarni klasifikator, klase se mogu označiti kao pozitivna i negativna. Za prethodno pomenuti primjer, problem detekcije bolesti na medicinskim slikama, slika koja prikazuje bolest pripada pozitivnoj klasi, dok slika koja ne prikazuje bolest pripada negativnoj klasi. Ukoliko je testni uzorak iz pozitivne klase tačno klasifikovan, označava se kao tačno pozitivan (eng. *true positive* - TP), a ukoliko je netačno klasifikovan, označava se kao lažno negativan (eng. *false negative* - FN). Analogno, ako je testni uzorak iz negativne klase tačno klasifikovan, označava se kao tačno negativan (eng. *true negative* - TN), a ukoliko je netačno klasifikovan, označava se kao lažno pozitivan (eng. *false positive* - FP). Svi ishodi se mogu prikazati tabelom 2.1. koja se još i naziva matrica konfuzije [10-11]:

Tabela 2.1. Matrica konfuzije

		Klasifikacija	
		Pozitivna	Negativna
Stvarna klasa	Pozitivna	TP	FN
	Negativna	FP	TN

Polazeći od date matrice konfuzije, može se izračunati više mjera efektivnosti klasifikatora. Tačnost klasifikatora se sada može izračunati kao [10]:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.2}$$

Udio tačno pozitivnih uzoraka (eng. *true positive rate* - TPR) je odnos broja tačno klasifikovanih pozitivnih uzoraka i ukupnog broja pozitivnih uzoraka u testnom skupu [10]:

$$TPR = \frac{TP}{TP + FN}. \tag{2.3}$$

Ova veličina se naziva i odziv (eng. *recall*) klasifikatora. Udio tačno negativnih uzoraka (eng. *true negative rate* - TNR) je odnos broja tačno klasifikovanih negativnih uzoraka i ukupnog broja negativnih uzoraka u testnom skupu [10]:

$$TNR = \frac{TN}{FP + TN}. \tag{2.4}$$

Udio lažno pozitivnih uzoraka (eng. *false positive rate* - FPR) je odnos broja pogrešno klasifikovanih negativnih uzoraka i ukupnog broja negativnih uzoraka u testnom skupu [10]:

$$FPR = \frac{FP}{FP + TN}. \tag{2.5}$$

Preciznost (eng. *precision*) klasifikatora se definiše kao odnos broja tačno klasifikovanih pozitivnih uzoraka i ukupnog broja uzoraka klasifikovanih u pozitivnu klasu [10]:

$$P = \frac{TP}{TP + FP}. \tag{2.6}$$

Često se koristi i F1 mjera koja je jednaka harmonijskoj sredini preciznosti i odziva [10]:

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{R + P}. \tag{2.7}$$

F1 mjera omogućava ravnotežu između ove dvije mjere, pružajući jedinstvenu vrijednost koja uzima u obzir oba aspekta performansi klasifikatora. To je posebno važno u kontekstu neuravnoteženih skupova podataka gdje jednostavno praćenje tačnosti može dovesti do pogrešnih zaključaka o stvarnoj efikasnosti modela [11].

2.2. Klasterizacija

Klasterizacija je postupak koji za cilj ima grupisanje sličnih podataka u grupe, klastere, tako da su elementi istog klastera „bliži“ jedni drugima nego što su elementima različitih klastera. Za razliku od nadgledanog obučavanja, klasterizacija je tipičan primjer nenadgledanog obučavanja (eng. *unsupervised learning*), gdje ne postoje unaprijed zadane oznake klasa.

Neka je skup podataka $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N}\}$, gdje $\boldsymbol{x_i} \in \mathbb{R}^d$. Cilj klasterizacije je da se ovaj skup podijeli u k disjunktnih podskupova (klastera) $C_1, C_2, \dots, C_K$, tako da važi:

$$\bigcup_{k=1}^{K} C_k = X \quad \text{i} \quad C_i \cap C_j = \emptyset \text{ za } i \neq j, \tag{2.8}$$

te da su podaci unutar klastera što je moguće sličniji jedni drugima, dok su podaci iz različitih klastera što različitiji.

Uobičajeno je da se definicija sličnosti ili distance zasniva na nekoj metodi mjerenja rastojanja, najčešće na euklidskom rastojanju, Menhetn rastojanju ili kosinusnoj sličnosti. Izbor odgovarajuće mjere zavisi od prirode podataka i korišćene metode klasterizacije.

K-means je jedna od najpoznatijih i najčešće korišćenih metoda klasterizacije, a zasniva se na minimizaciji unutar-klasterske sume kvadrata rastojanja. Ključni koraci algoritma su [10]:

1. Izabrati k početnih centroida $\mu_1, \mu_2, \dots, \mu_k$.
2. Svakom podatku x_i dodeliti klaster čiji je centorid najbliži (po izabranoj metrici, obično euklidskoj).
3. Za svaki klaster C_j izračunati novi centorid:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i. \tag{2.9}$$

4. Ponavljati korake (2) i (3) dok se centoridi ne ustale, dok se funkcija greške ne promijeni ispod zadatog praga, ili se dosegne maksimalan dozvoljen broj iteracija.

Glavna optimizaciona funkcija koju *K-means* minimizuje je:

$$J = \sum_{j=1}^{k} \sum_{x \in C_j} |x - \mu_j|_2^2. \tag{2.10}$$

Postoje različite primjene za klasterizaciju, kao što je grupisanje korisnika prema sličnim osobinama, grupisanje proizvoda na osnovu opisa tih proizvoda itd.

2.3. Dotreniranje

U mašinskom učenju, posebno kada se koriste duboke neuronske mreže, treniranje od nule (eng. *training from scratch*) često je izuzetno zahtjevno, kako računarski tako i u pogledu podataka. Dotreniranje predstavlja rješenje kojim se postojeći, već istrenirani model, najčešće na velikoj i raznovrsnoj bazi podataka, prilagođava novom, često manjem ili specifičnijem domenu. Na taj način se čuva "znanje" koje je mreža već stekla, a ujedno se optimizuje za novi zadatak ili novi skup podataka. Neka je:

- $\mathcal{D}_{veliki}$: veliki skup podataka (npr. *ImageNet*, COCO, *Wikipedia* tekst),
- $\mathcal{D}_{\text{novi}}$: novi, manji ili specifičniji skup podataka (npr. slike odjevnih predmeta).

I neka je data mreža sa parametrima θ. Nakon početnog treninga na $\mathcal{D}_{\text{veliki}}$, se dobije pretrenirani model θ_0. U procesu dotreniranja, se traže novi parametri θ^* koji minimizuju funkciju cijene $\mathcal{L}$ na $\mathcal{D}_{\text{novi}}$, ali počevši od početne vrijednosti θ_0. Formalno:

$$\theta^* = \arg\min_{\theta} [\, \mathcal{L}(f(\cdot;\theta), \mathcal{D}_{\text{nv}})] \quad \text{sa inicijalizacijom} \quad \theta \leftarrow \theta_0.$$

Funkcija cijene $\mathcal{L}$ može biti npr. kros-entropija za klasifikacijski zadatak, ili neki drugi prikladan kriterijum.

Postoji više varijanti dotreniranja:

1. Zamrzavanje (eng. *freezing*) ranih slojeva: Rani slojevi modela služe kao detektori opštih obilježja, pa se oni mogu ostaviti zamrznuti. Treniraju se samo kasniji slojevi.
2. Potpuno dotreniranje: Svi slojevi su trenirani uz manju brzinu učenja, jer treba izbjeći prevelike promjene parametara koji su već prethodno naučeni.
3. Dodavanje novih slojeva: Ponekad se dodaju dodatni slojevi ili glava (eng. *head*) modela, koji se treniraju na specifičnom zadatku, dok su osnovni slojevi (tzv. *backbone*) manje-više zamrznuti.

3. NEURONSKE MREŽE

Neuronska mreža je ML model inspirisan strukturom i funkcionisanjem ljudskog mozga. Sastoji se od slojeva neurona, od kojih svaki neuron prima više ulaza, obrađuje te informacije i generiše izlaz. Ako se na ulaz neurona dovode signali $x_1, x_2, \dots, x_N$, tada je signal na njegovom izlazu dat izrazom

$$a = g(z), \tag{3.1}$$

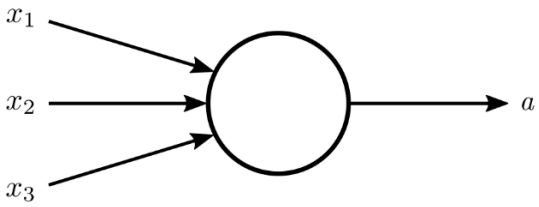
gdje je $g(\cdot)$ aktivaciona funkcija neurona, a z je afina funkcija ulaznih signala

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_N x_N + b. \tag{3.2}$$

Veličine $w_1, w_2, \dots, w_N$ su težine ulaza neurona, a b je ofset. Težine određuju koliki uticaj svaki ulazni signal ima na izlaz neurona. Kako bi se dobio kompaktniji izraz, često se smatra da neuron ima još jedan ulaz x_0 čija je vrijednost uvijek jednaka jedinici, a njegova težina w_0, jednaka je ofsetu. Sada je

$$z = w_0 x_0 + w_1 x_1 + \cdots + w_N x_N = \boldsymbol{w^T x}, \tag{3.3}$$

gdje je $w = [w_0, w_1, \dots, w_N]^T$ i $x = [x_0, x_1, \dots, x_N]^T$. U ovom obliku, z predstavlja linearnu kombinaciju ulaznih signala. Na slici 3.1 je dat šematski prikaz jednog neurona.



Slika 3.1: Šematski prikaz jednog neurona [10]

Aktivaciona funkcija $g(z)$ je nelinearna funkcija koja određuje izlaz neurona na osnovu afine funkcije z. Istorijski, prvobitni pokušaji korištenja neuronskih mreža bez nelinearnih aktivacionih funkcija pokazali su se neuspješnim za složene zadatke. Matematički, ako se koriste linearne funkcije kroz sve slojeve mreže, konačni izlaz bi bio linearna kombinacija ulaznih signala, čime bi se cijela mreža mogla svesti na jedan jedini neuron koji primjenjuje linearnu transformaciju. Ova realizacija dovela je do uvođenja nelinearnih aktivacionih funkcija, pomoću kojih su neuronske mreže postale univerzalni aproksimatori funkcija, sposobni da nauče širok spektar složenih obrazaca i odnosa u podacima.

Istorijski, prva predložena aktivaciona funkcija imala je oblik odskočne funkcije [10]:

$$g(z) = \begin{cases} 0, & \text{ako je } z \leq 0 \\ 1, & \text{ako je } z > 0 \end{cases}, \tag{3.4}$$

koja je bila nepraktična jer nije diferencijabilna, pa su vremenom u upotrebu ušle i druge vrste aktivacionih funkcija kao što su sigmoidna, hiperbolički tangens i ispravljačka aktivaciona funkcija (eng. *rectified linear unit* - ReLU) koja se pokazala kao dobar izbor za probleme klasifikacije. Data je izrazom:

$$g(z) = \max(0, z). \tag{3.5}$$

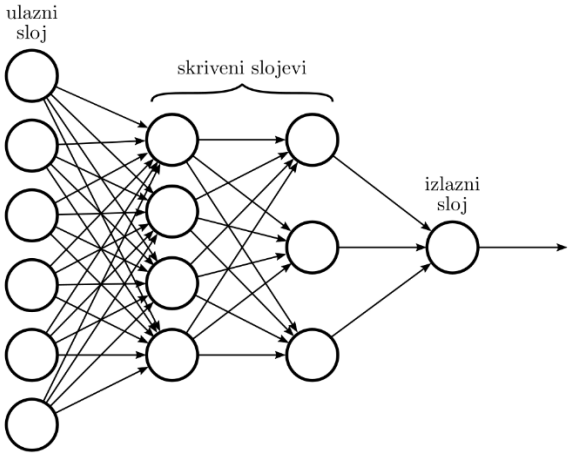
Mana ReLU leži u tzv. problemu "umirućih neurona", gdje neuroni sa negativnim vrijednostima afine funkcije nikada ne bivaju aktivirani, ali i zbog toga što je izlaz aktivacione funkcije nula za različite negativne vrijednosti afine funkcije, čime se ignorišu nijanse, pa su razvijene različite generalizacije za ReLU. Najpoznatija generalizacija je probojni ReLU:

$$g(z) = \begin{cases} \alpha z, & \text{ako je } z \leq 0 \\ z, & \text{ako je } z > 0 \end{cases}, \tag{3.6}$$

koji dopušta prolazak malih negativnih vrijednosti, gdje je α obično neka mala vrijednost i predstavlja hiperparametar.

Pojedinačni neuroni posjeduju ograničenu računarsku sposobnost, ali kada se veliki broj njih međusobno poveže, formira se model koji je sposoban da efikasno obrađuje kompleksne obrasce i informacije. Neuroni se organizuju u slojeve, a način organizacije, broj slojeva i način na koji su povezani čine arhitekturu mreže. Najjednostavnija, u praksi primjenljiva, arhitektura neuronske mreže je višeslojna mreža bez povratnih veza (eng. *feed-forward network* - FFNN). U ovoj mreži, neuroni iz jednog sloja mogu biti povezani samo sa neuronima iz sljedećeg sloja. Takođe, ne postoje veze između neurona iz istog sloja, veze koje preskaču slojeve, kao ni povratne veze.

Prvi sloj neuronske mreže naziva se ulaznim slojem, posljednji se naziva izlaznim slojem, dok se svi ostali slojevi nazivaju skrivenim slojevima. Neuroni u ulaznom sloju imaju samo jedan ulaz i jedan izlaz, pri čemu je izlazni signal jednak ulaznom. Na taj način, ulazni neuroni ne obrađuju signal, već služe za konzistentnu notaciju ulaznih i izlaznih signala mreže. Neuroni u skrivenim slojevima primaju ulazne signale iz izlaza neurona prethodnog sloja i obrađuju ih, stvarajući izlazne signale koji se prosljeđuju narednim slojevima. Ovi signali se hijerarhijski transformišu kroz mrežu, omogućavajući izdvajanje karakteristika ulaznih podataka na različitim nivoima apstrakcije. Na kraju, neuroni u izlaznom sloju generišu izlazne signale mreže, koji predstavljaju konačni rezultat obrade. Na slici 3.2 je dat šematski prikaz arhitekture jedne jednostavne neuronske mreže.



Slika 3.2: Šematski prikaz neuronske mreže [10]

Broj slojeva u neuronskoj mreži varira u zavisnosti od složenosti problema. Osnovna mreža ima tri sloja: ulazni, jedan skriveni i izlazni sloj. Povećanjem broja skrivenih slojeva, mreža može da uči složenije obrasce i apstraktne karakteristike podataka, čime se povećava njena

sposobnost da rješava kompleksnije probleme. Broj skrivenih slojeva, zajedno sa brojem neurona u svakom sloju, predstavlja hiperparametre mreže. Određivanje optimalne arhitekture mreže može se vršiti različitim pristupima.

- Najčešće se koristi unakrsna validacija koja uključuje testiranje različitih kombinacija hiperparametara kako bi se pronašla najbolja konfiguracija.
- Heuristički pristupi koriste postojeće znanje i literaturu za kreiranje mreža sa dokazanim učinkom, ali zahtijevaju ekspertizu i mogu biti vremenski zahtijevni.
- U posljednje vrijeme, sve više se koristi automatizovani pristup za određivanje arhitekture mreže, tzv. pretraga neuronskih arhitektura (eng. *neural architecture search* - NAS). NAS primjenjuje tehnike pretrage koje se često baziraju na evolutivnim algoritmima, kao što je genetski algoritam. Ovi algoritmi automatski istražuju prostor mogućih arhitektura i optimizuju mrežu za specifične zadatke, čime se smanjuje potreba za ručnim podešavanjem i ekspertskim znanjem, i omogućava brža i efikasnija optimizacija mrežnih arhitektura.

U mreži sa L slojeva, gdje posljednji sloj sadrži $n(L)$ neurona, za k-ti neuron l-tog sloja, afina funkcija je data izrazom:

$$z_k^{(l)} = \sum_{j=0}^{n(l-1)} \omega_{kj}^{(l-1)} a_j^{(l-1)}, \tag{3.7}$$

gdje $\omega_{kj}^{(l-1)}$ predstavlja težinu veze od j-tog neurona u $(l-1)$-om sloju ka k-tom neuronu u l-tom sloju, dok je $a_j^{(l-1)}$ izlaz iz j-tog neurona u prethodnom sloju $(l-1)$. Težina $\omega_{k0}^{(l-1)}$, predstavlja ofset. Broj $n(l-1)$ označava ukupan broj neurona u $(l-1)$-om sloju.

Vektor težina za k-ti neuron posljednjeg sloja može se predstaviti kao $\boldsymbol{w_k^{(L-1)}}$, a vektor izlaza iz pretposljednjeg sloja kao $\boldsymbol{a^{(L-1)}}$. Vektor izlaza za neurone posljednjeg sloja, prije primjene aktivacione funkcije, može se izraziti kao:

$$\boldsymbol{z_k^{(L)}} = \left(\boldsymbol{w_k^{(L-1)}}\right)^T \boldsymbol{a^{(L-1)}} \text{ za } k = 1, \dots, n^{(L)}. \tag{3.8}$$

Za problem klasifikacije uzoraka, broj neurona u izlaznom sloju mreže je jednak broju klasa u koje se uzorci klasifikuju. Izlaz pretposljednjeg sloja, dat sa prethodnim izrazom, čine logiti, skup numeričkih vrijednosti koji ne govore direktno o pripadnosti klasi. Zbog toga izlazni signali prolaze kroz softmaks aktivacionu funkciju:

$$a_k^{(L)} = g\left(z_1^{(L)}, \dots, z_{n(L)}^{(L)}\right) = \frac{e^{z_k^{(L)}}}{\sum_{i=1}^{n(L)} e^{z_i^{(L)}}} \text{ za } k = 1, \dots, n^{(L)}. \tag{3.9}$$

Softmaks funkcija transformiše logite u vjerovatnoće tako da je zbir svih vjerovatnoća jednak 1. Na ovaj način, omogućava mreži da generiše vjerovatnoće za svaku klasu, čineći je korisnom za zadatke višeklasne klasifikacije.

Vrijednost izraza predstavlja procijenjenu vjerovatnoću da uzorak pripada k-toj klasi. Uzorak se klasifikuje u onu klasu koja ima maksimalnu procijenjenu vjerovatnoću.

Prilikom obučavanja neuronskih mreža za klasifikaciju, trening uzorci se predstavljaju kao parovi (x_i, y_i), gdje $x_i \in \mathbb{R}^N$ označava vektor obilježja, a y_i je oznaka klase kojoj uzorak pripada.

U situacijama kada neuronska mreža koristi softmaks funkciju u izlaznom sloju, obuka modela se ostvaruje minimizacijom funkcije cijene zasnovane na kategorijskoj krosentropiji. Funkcija cijene data je izrazom:

$$J_0(W) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n^{(L)}} t_{i,k} \ln a_k^{(L)}, \tag{3.10}$$

gdje je n broj trening uzoraka, $n^{(L)} = K$ broj neurona u izlaznom sloju, a W vektor težina svih neurona u mreži. Vrijednost $t_{i,k} = 1$ ako uzorak pripada klasi c_k, odnosno $y_i = c_k$, dok je $t_{i,j} = 0$ za sve ostale klase $j \neq k$. Da bi se riješio problem preprilagođavanja, funkciji cijene se dodaje regularizacioni član. Pa funkcija cijene dobija oblik:

$$J(W) = J_0(W) + \lambda R(W), \tag{3.11}$$

gdje λ predstavlja hiperparametar koji kontroliše uticaj regularizacionog člana $R(W)$ u odnosu na osnovnu funkciju cijene $J_0(W)$. Regularizacija time sprječava da težine postanu prevelike, što bi moglo dovesti do prekomjernog prilagođavanja modela trening podacima.

$L1$ regularizacija je metoda koja koristi apsolutne vrijednosti težina kao kaznu. Kada se doda regularizacioni član u funkciju cijene, $L1$ penalizuje veće težine, te neke od njih postanu nula. Ovim se smanjuje broj aktivnih težina, zadržavajući samo one karakteristike koje su najvažnije za predikciju. $L1$ regularizacija je posebno korisna kada se radi s podacima koji sadrže veliki broj irelevantnih ili redundantnih karakteristika. Na taj način se postiže model koji je jednostavniji, brži za treniranje i lakši za interpretaciju, jer uklanja nebitne informacije.

$L2$ regularizacija, poznata i kao *Ridge* regularizacija, koristi drugačiji pristup. Umjesto da postavlja težine na nulu, kao što to radi $L1$, $L2$ penalizuje kvadrat težina. Ova tehnika smanjuje sve težine proporcionalno, ali rijetko uklanja bilo koju od njih u potpunosti. Umjesto toga, model održava sve karakteristike, ali ih čini manje izraženim. $L2$ regularizacija je korisna kada su sve karakteristike važne za predikciju, ali je potrebno smanjiti njihovu složenost kako bi model bio „glatkiji“[2] i otporniji na preprilagođavanje. Kombinacija ovih pristupa, poznata kao *Elastic Net* regularizacija, koristi prednosti obje - $L1$ regularizacija eliminiše nevažne karakteristike, dok $L2$ održava „glatkoću“ modela i smanjuje težine bez njihovog potpunog uklanjanja. *Elastic Net* je koristan u situacijama kada postoji mnogo karakteristika, od kojih su neke važne, ali ne sve, te je potreban balans između uklanjanja nebitnih i zadržavanja ključnih informacija.

[2] Glatkoća modela (eng. *smoothness*) označava osobinu modela da ne pravi nagle promjene u izlazu usljed malih promjena u ulaznim podacima.

Minimizacija funkcije cijene u neuronskim mrežama, predstavlja zadatak koji se ne može riješiti analitičkim putem, jer je u pitanju nelinearna funkcija sa velikim brojem parametara [10]. Zbog toga se koriste iterativne metode, pri čemu je gradijentni spust najčešće korišćeni algoritam. Gradijentni spust ažurira težine W na osnovu gradijenta funkcije cijene u odnosu na te težine. Taj gradijent predstavlja smjer najbržeg porasta funkcije greške, pa se težine ažuriraju u suprotnom smjeru od gradijenta kako bi se greška smanjila. Proces se odvija iterativno, gdje se u svakoj iteraciji težine koriguju za iznos koji je proporcionalan gradijentu i hiperparametru poznatom kao stopa učenja η:

$$W \leftarrow W - \eta \nabla J(W), \tag{3.12}$$

gdje $\nabla J(W)$ predstavlja gradijent funkcije cijene u odnosu na težine. Stopa učenja η određuje veličinu koraka koje model pravi u smjeru minimizacije funkcije cijene. Ako je η prevelika, model može preskočiti minimum, dok premala vrijednost η može dovesti do sporog konvergiranja.

S obzirom na to da neuronske mreže sadrže mnogo slojeva, svaki sa svojim težinama, proces minimizacije funkcije greške se odvija kroz cijelu mrežu. Svaki sloj mreže transformiše ulazne podatke u odgovarajuće izlazne vrijednosti koje se prosljeđuju dalje kroz mrežu, dok izlazni sloj daje konačnu predikciju. Greška se prvo računa na izlazu mreže, jer je tada poznata stvarna vrijednost y i predikcija modela $\hat{y}$. Ta greška se zatim koristi za računanje gradijenata težina u svim prethodnim slojevima, sloj po sloj, unazad ka početku mreže. Ovaj postupak je poznat kao propagacija unazad (eng. *backpropagation*) [10-11].

Propagacija unazad se zasniva na lančanom pravilu diferenciranja, koje omogućava efikasno računanje derivacija složenih funkcija. U neuronskim mrežama sa višestrukim slojevima, gdje je izlaz funkcija ulaza kroz niz slojeva transformacija, lančano pravilo omogućava da se računa kako promjena težina u svakom sloju utiče na grešku na izlazu mreže. Lančano pravilo navodi da, ako postoji složena funkcija $f\big(g(x)\big)$, derivacija ove funkcije u odnosu na x je data kao:

$$\frac{d}{dx} f\big(g(x)\big) = f'\big(g(x)\big) \cdot g'(x). \tag{3.13}$$

U neuronskim mrežama, aktivacija svakog sloja $a^{(l)}$ zavisi od prethodnog sloja, tako da promjene u težinama sloja $W^{(l)}$ utiču na krajnju grešku kroz niz međusobno povezanih transformacija. Primjenom lančanog pravila, moguće je pratiti kako promjena težina u svakom sloju doprinosi promjeni ukupne greške. Na primjer, derivacija funkcije greške J u odnosu na težine u sloju l može se izraziti kao:

$$\frac{\partial J}{\partial W^{(l)}} = \frac{\partial J}{\partial a^{(l+1)}} \cdot \frac{\partial a^{(l+1)}}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial W^{(l)}}. \tag{3.14}$$

Ovaj izraz pokazuje kako se gradijenti funkcije greške propagiraju unazad kroz mrežu, od izlaznog sloja ka ulaznom sloju, sloj po sloj.

Razlog zbog kojeg se gradijenti računaju unazad, a ne unaprijed, leži u efikasnosti. Na izlazu mreže direktno postoje informacije o grešci, jer je tada poznata razlika između predviđanja i stvarne vrijednosti. Da bi se gradijenti u ranijim slojevima izračunali, potrebno

je prvo znati kako greška na izlazu zavisi od tih slojeva. Propagacija unazad omogućava da se, počevši od greške na izlazu, gradijenti prenesu unazad kroz mrežu koristeći lančano pravilo, čime se efikasno računa kako svaka težina doprinosi ukupnoj grešci. Ako bi se gradijenti računali unaprijed, to bi zahtijevalo mnogo više proračuna i memorije, jer bi bilo potrebno prvo računati uticaj svakog sloja unaprijed, što bi bilo nepraktično u složenim dubokim mrežama.

Iako su se FFNN uspješno pokazale u mnogim problemima iz oblasti ML, one imaju poteškoća u radu sa određenim vrstama podataka, zbog čega su razvijene specifične arhitekture kao što su konvolucione neuronske mreže i rekurentne neuronske mreže.

3.1. Konvolucione neuronske mreže

FFNN nisu prilagođene za rad sa (rasterskim) slikama [10]. Ako je dat ulazni podatak x dimenzija $\mathrm{H} \times \mathrm{W} \times \mathrm{C}$, onda bi ulazni sloj FFNN mreže imao $H \cdot W \cdot C$ neurona, npr. za sliku dimenzija 256×256 piksela s tri kanala (RGB) mreža ima $256 \times 256 \times 3 = 196.608$ ulaznih neurona. Kako je svaki neuron u prvom skrivenom sloju je povezan sa svih 196.608 ulaznih neurona, to stvara ogroman broj težinskih parametara već u prvom skrivenom sloju. Uzimajući u obzir da tipična mreža ima više skrivenih slojeva, broj parametara eksponencijalno raste, čineći obučavanje ovakvih mreža izrazito računarski zahtjevnim. Uz to postoji i značajan rizik od preprilagođavanja ulaznim podacima.

Slike imaju izraženu prostornu strukturu, pa su veze između bliskih piksela važnije od veza između udaljenih piksela. Ova osobina je inspirisala razvoj CNN arhitekture, gdje su neuroni u skrivenom sloju povezani samo sa ulaznim neuronima prostorno bliskih piksela. Ova lokalna povezanost znači da je neuron u skrivenom sloju povezan samo sa pikselima unutar svog receptivnog polja - dijela slike koji utiče na njegovu aktivaciju.

Svaki neuron u jednom sloju CNN koristi isti skup težina za sve pozicije unutar svog receptivnog polja, a taj skup težina se naziva filter. Filter se "pomjera" ili "klizi" po slici, što odgovara matematičkoj operaciji konvolucije, pa se filter često naziva i konvolucioni kernel. Formalno, dvodimenzionalna konvolucija slike f i kernela h, veličine $(2a + 1) \times (2b + 1)$, se računa kao [10]:

$$g(i,j) = \sum_{k=-a}^{a} \sum_{l=-b}^{b} h(k,l) f(i-k, j-l). \tag{3.15}$$

Filteri sa neparnim dimenzijama, kao što su 3×3 ili 5×5, se koriste kako bi postojao jasno definisan centralni piksel. To olakšava poravnanje filtera sa svakim pikselom u slici jer centralna tačka filtera može biti precizno poravnata sa trenutnim pikselom u slici. Međutim, pri primjeni konvolucije bez dodatnih koraka, dimenzije rezultujuće mape obilježja smanjuju se u odnosu na ulaznu sliku. Ako je ulazna slika dimenzija $W \times H$, a konvolucioni filter dimenzija $(2a + 1) \times (2b + 1)$, dimenzije izlazne mape obilježja će biti $(W - 2a) \times (H - 2b)$. Ovo smanjenje dimenzija može predstavljati problem, posebno kod dubokih mreža sa više konvolucionih slojeva, jer se informacija gubi na rubovima slike sa svakom konvolucijom. Opcionalno da bi se očuvale dimenzije ulazne slike nakon konvolucije i zadržali rubne informacije, koristi se tehnika koja se naziva popunjavanje (eng. *padding*). Popunjavanje je proces dodavanja dodatnih redova oko rubova slike. Najčešće se koristi popunjavanje nulama.

Konvolucija omogućava filteru da detektuje specifična obilježja slike, kao što su ivice, teksture ili oblici, nezavisno od njihove tačne pozicije na slici. Kao rezultat primjene filtera na cijelu sliku, dobija se mapa obilježja (eng. *feature map*), koja prikazuje gdje se određeno obilježje pojavljuje na slici. Da bi mreža prepoznala raznolike obrasce, koristi se više različitih filtera u svakom konvolucionom sloju. Na početku, ulazna slika obično ima jedan kanal, ako je crno-bijela, ili tri kanala, ako je u boji, tj. RGB slika. Međutim, prolaskom kroz slojeve CNN-a, broj kanala se povećava. To se dešava zato što svaki konvolucioni sloj koristi više različitih filtera za detekciju različitih osobina slike, a svaka mapa obilježja predstavlja jedan kanal u izlazu tog sloja.

Ako CNN radi sa ulaznim slikama dimenzija $256 \times 256 \times 3$, kao u prethodnom primjeru za FFNN i ako prvi konvolucioni sloj koristi npr. 32 filtera dimenzija $3 \times 3 \times 3$. Broj parametara u ovom sloju iznosi:

$$3 \times 3 \times 3 \times 32 = 864 \text{ parametara.} \tag{3.16}$$

Ako naredni konvolucioni sloj sadrži 64 filtera dimenzija $3 \times 3 \times 32$, broj parametara u tom sloju bi bio:

$$3 \times 3 \times 32 \times 64 = 18.432 \text{ parametara.} \tag{3.17}$$

Što je značajno manji broj parametara nego kod FFNN. Zbog toliko manjeg broja parametara mnogo je manji rizik od preprilagođavanja, ali i od problema pri treniranju kao što su problem „nestajućih“ gradijenata[3] ili problem „eksplozije“ gradijenata[4], a koji se pojavljuju kod FFNN i predstavljaju ozbiljan problem kod RNN o kojima će biti riječi kasnije.

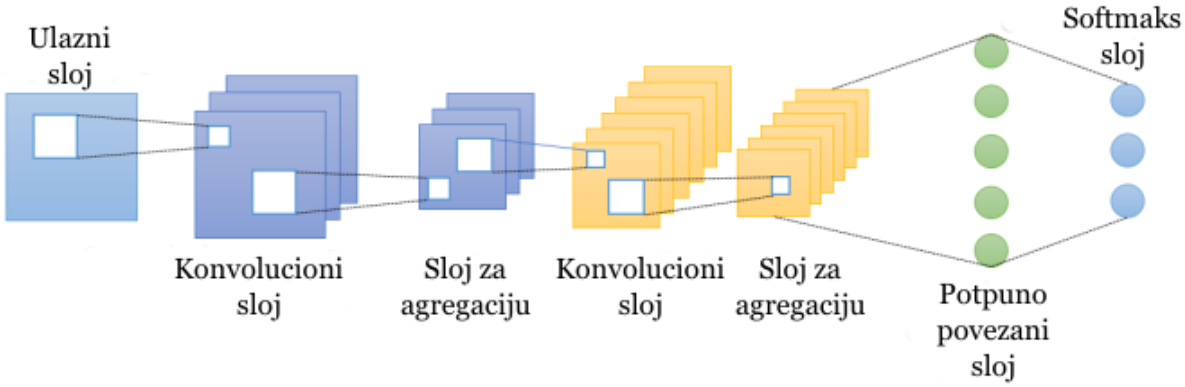
Problem sa konvolucionim filterima jeste prostorna osjetljivost. Npr., ako je filter naučio prepoznati slovo "L" na određenoj poziciji, čak i minimalni pomak ili promjena u rasporedu piksela može dovesti do toga da filter ne prepozna željeni obrazac. Ovaj problem proizlazi iz činjenice da konvolucija nije translacijski invarijantna. U stvarnim situacijama, objekti na slikama mogu se pojaviti na različitim mjestima, skalama ili orijentacijama, te je važno da model bude robustan na takve varijacije [11].

Kako bi se uvela translacijska invarijantnost vrši se pododmjeravanje (eng. *pooling*) mape obilježja. Pododmjeravanjem se agregiraju lokalne informacije, tako što se uzima najveća ili srednja vrijednost iz receptivnog polja. Time se omogućava da filter koji prepoznaje određenu značajku "reaguje" na tu značajku bez obzira na njenu preciznu lokaciju unutar određenog područja. Sloj dobijen na ovaj način naziva se sloj za agregaciju (eng. *pooling layer*). Na ovaj način se ujedno i smanjuje broj parametara mreže, a bez da to ima uticaj na rezultate obučavanja, te se i dodatno smanjuje mogućnost preprilagođavanja.

Kombinovanjem konvolucionih slojeva i slojeva za agregaciju formira se CNN. Međutim moguće je smanjiti broj slojeva i uštediti na vremenu za procesiranje podataka tako što se umjesto zasebnih slojeva za konvoluciju i agregaciju vrši "ugradnja" pododmjeravanja u sloj

[3] Problem gdje se gradijenti tokom propagacije unazad smanjuju do toliko male vrijednosti da model ne može efikasno učiti, što usporava ili potpuno zaustavlja proces treniranja.
[4] Suprotan problem od „nestajućih“ gradijenata, gdje se gradijenti tokom propagacije unazad povećavaju eksponencijalno, što može uzrokovati velike oscilacije u težinama i time uništiti stabilnost modela, što otežava ili onemogućava učenje.

za konvoluciju. Obično filter prelazi preko čitave slike, "klizeći" preko nje piksel po piksel, ali ako se pomak (eng. *stride*) poveća, tako da se neki pikseli preskaču (npr. svaki drugi), tada se dobija nova mapa obilježja čije su dimenzije umanjene u odnosu na ulazne. Iako se mapa obilježja dobijena na ovakav način razlikuje od one koja bi se dobila primjenom dva zasebna sloja, istraživanja su pokazala da su performanse gotovo iste u oba slučaja [11]. Konačno posljednjih par skrivenih slojeva u CNN su potpuno povezani slojevi, nakon kojih obično slijedi softmaks sloj kao izlazni sloj. Na slici 3.3 je dat šematski prikaz arhitekture CNN.



Slika 3.3: Šematski prikaz arhitekture CNN [15]

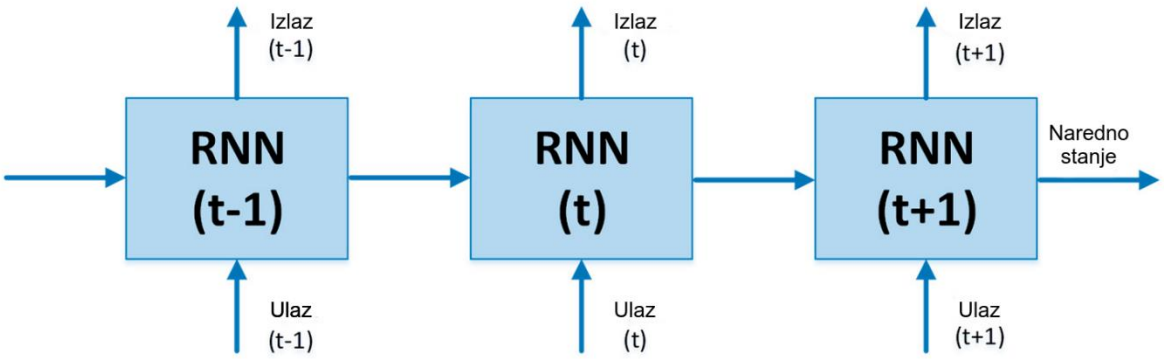
3.2. Rekurentne neuronske mreže

Sekvencijalni podaci su vrsta podataka u kojima redoslijed elemenata igra ključnu ulogu. Ovi podaci mogu biti vremenske serije, prirodni jezik, genetski nizovi, ili bilo koji drugi podaci gdje redoslijed unosa nosi informaciju o njihovoj međuzavisnosti. Upravo zbog te prirode podataka, tradicionalni ML modeli, koji pretpostavljaju nezavisnost između uzoraka, nisu prikladni za njihovu obradu.

Jedan od osnovnih izazova u obradi sekvencijalnih podataka je modelovanje zavisnosti između elemenata u sekvenci. Na primjer, u zadacima obrade jezika značenje riječi zavisi od prethodnih riječi u rečenici. Ova zavisnost može biti kratkoročna, kada je važno samo nekoliko prethodnih elemenata, ali često je dugoročna, kada je potrebno uzeti u obzir cijelu sekvencu ili veći dio nje.

Pored toga, još jedan izazov je varijabilnost dužine sekvenci, što otežava njihovu obradu standardnim metodama koje zahtijevaju fiksne ulazne dimenzije. Na primjer, rečenice u prirodnom jeziku mogu imati različit broj riječi, dok vremenske serije mogu trajati različit broj vremenskih koraka.

Rekurentne neuronske mreže (eng. *recurrent neural networks* - RNN) su dizajnirane za obradu sekvencijalnih podataka jer obrađuju podatke redom, pri čemu svaka nova informacija zavisi od prethodno obrađenih elemenata sekvence. Za razliku od FFNN, koje obrađuju ulaze nezavisno, i CNN, koje modeluju lokalne zavisnosti u podacima, RNN modeli koriste unutrašnje skriveno stanje koje se dinamički ažurira pri svakom koraku sekvence. Ovaj mehanizam omogućava RNN modelima da zadrže informacije o prethodnim elementima u sekvenci i koriste ih kao kontekst za donošenje odluka o trenutnom elementu. Na slici 3.4 je dat šematski prikaz arhitekture RNN.



Slika 3.4: Šematski prikaz arhitekture RNN [16]

RNN funkcioniše tako što na svakom vremenskom koraku t, uzima ulaz x_t i ažurira skriveno stanje h_t na osnovu prethodnog stanja h_{t-1} i trenutnog ulaza:

$$h_t = f(W_h h_{t-1} + W_x x_t + b), \tag{3.18}$$

gdje su W_h i W_x težinski parametri, b je pomjeraj (eng. *bias*), a f je nelinearna aktivaciona funkcija. Pa je izlaz mreže u vremenskom koraku t dat sa izrazom:

$$y_t = g\big(W_y h_t + c\big), \tag{3.19}$$

gdje su W_y težinski parametar za mapiranje skrivenog stanja h_t u izlazni vektor y_t, c pomjeraj za izlaz i g funkcija za izračunavanje izlaza. U zavisnosti od zadatka, g može biti nelinearna funkcija kao što je softmaks (za klasifikaciju) ili linearna funkcija (za regresiju). Ovaj proces predstavlja prosljeđivanje unaprijed (eng. *forward pass*) kroz mrežu.

Treniranje RNN se vrši pomoću propagacije unazad kroz vrijeme (eng. *backpropagation through time*), gdje se uzima u obzir i vremenska dimenzija [11]. Sam proces treniranja se može podijeliti u tri faze. Prvo se vrši prosljeđivanje unaprijed koje je prethodno opisano. Nakon toga se računa ukupna greška na izlazu mreže L, koja je data izrazom:

$$L = \sum_{t=1}^{T} L_t = \sum_{t=1}^{T} \mathrm{l}(y_t, \hat{y}_t), \tag{3.20}$$

gdje je $\mathrm{l}(y_t, \hat{y}_t)$ funkcija cijene (npr. unakrsna entropija za klasifikaciju ili srednja kvadratna greška za regresiju), y_t predviđeni izlaz, $\hat{y}_t$ stvarna vrijednost u vremenskom koraku t. Konačno se greška prosljeđuje unazad (kroz vrijeme) od posljednjeg koraka T do prvog koraka $t = 1$. Na svakom koraku, gradijenti se izračunavaju u odnosu na težinske parametre W_h, W_x i W_y, i koriste se za ažuriranje ovih parametara. Gradijent ukupne greške kroz vrijeme je dat izrazom:

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial W}, \tag{3.21}$$

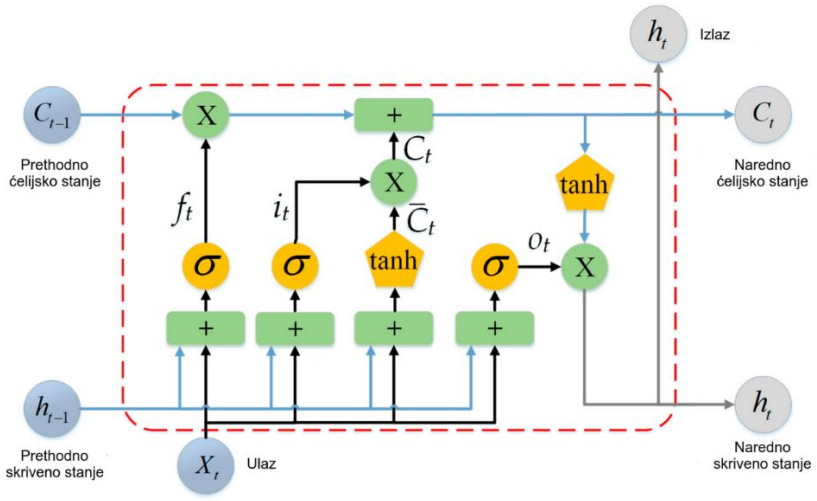
gdje W označava bilo koji od težinskih parametara $\big(W_h, W_x, W_y\big)$.

Postoje određene poteškoće u treniranju RNN kao što je problem nestajućeg gradijenata koji se dešava kada informacije iz ranijih vremenskih koraka postaju sve manje uticajne tokom procesa treniranja. To znači da model teško uči dugoročne zavisnosti jer su promjene u težinskim parametrima, koje su potrebne da bi se informacije prenijele iz daleke prošlosti, zanemarljivo male. Kao rezultat, RNN može "zaboraviti" važne informacije iz ranih dijelova sekvence, što utiče na tačnost predviđanja. Nasuprot tome, problem eksplodirajućeg gradijenta nastaje kada gradijenti tokom treniranja postaju izrazito veliki. Ovo može dovesti do nestabilnosti u mreži, gdje se težinski parametri ažuriraju prevelikim vrijednostima, što može uzrokovati lošu konvergenciju ili čak divergirajuće rezultate. Oba ova problema su posebno izražena kod dužih sekvenci, gdje informacije moraju da putuju kroz mnoge vremenske korake.

Da bi se djelimično prevazišli ovi problemi, razvijeni su napredniji modeli RNN kao što su duga kratkoročna memorija (eng. *long short-term memory* - LSTM) i rekurentna jedinica sa vratima (eng. *gated recurrent unit* - GRU) [11]. Ove arhitekture uvode mehanizme poznate kao vrata (eng. *gates*) koji kontrolišu protok informacija unutar mreže i omogućavaju modelu da zadrži relevantne informacije kroz duže vremenske periode i da efikasnije rukuje dugoročnim zavisnostima.

3.2.1. LSTM

LSTM uvodi tri tipa vrata: vrata zaboravljanja (eng. *forget gate*), ulazna vrata (eng. *input gate*) i izlazna vrata (eng. *output gate*). Ova vrata pomažu u odabiru koje informacije će biti dodate, koje će biti sačuvane i koje će biti odbačene iz mreže. Pored vrata, LSTM uvodi i C_t, ćelijsko stanje (eng. *cell state*) koje predstavlja memorijski lanac kroz vremenske korake, pomoću kog se čuvaju dugoročne zavisnosti. Na slici 3.5 je dat šematski prikaz arhitekture LSTM.



Slika 3.5: Šematski prikaz arhitekture LSTM [16]

Vrata zaboravljanja odlučuju koje informacije iz prethodnog ćelijskog stanja C_{t-1} treba zaboraviti:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big), \tag{3.22}$$

gdje je W_f matrica težina zaboravnih vrata, h_{t-1} je prethodno skriveno stanje, x_t je trenutni ulaz, $[h_{t-1}, x_t]$ konkatenacija vektora, b_f je pomjeraj za zaboravna vrata i σ sigmoidna funkcija kojom se niveliše važnost informacija.

Ulazna vrata odlučuju koje nove informacije će se dodati u ćelijsko stanje kroz novo kandidatsko stanje:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{3.23}$$

gdje je W_i matrica težina ulaznih vrata i b_i je pomjeraj ulaznih vrata.

Na ulaznim vratima se računa kandidat za novo ćelijsko stanje $\widetilde{C_t}$ koji se normalizuje u opsegu od -1 do 1, korištenjem hiperboličkog tangensa:

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{3.24}$$

gdje je W_C matrica težine za generisanje kandidata ćelijskog stanja, b_C je pomjeraj za kandidat ćelijskog stanja.

Trenutno, odnosno ažurirano ćelijsko stanje C_t je linearna kombinacija prethodnog stanja, modifikovanog zaboravnim vratima, i novog kandidata, skaliranog ulaznim vratima:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}. \tag{3.25}$$

Na izlaznim vratima se odlučuje koji dio ažuriranog ćelijskog stanja će uticati na izlaz skrivenog stanja.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{3.26}$$

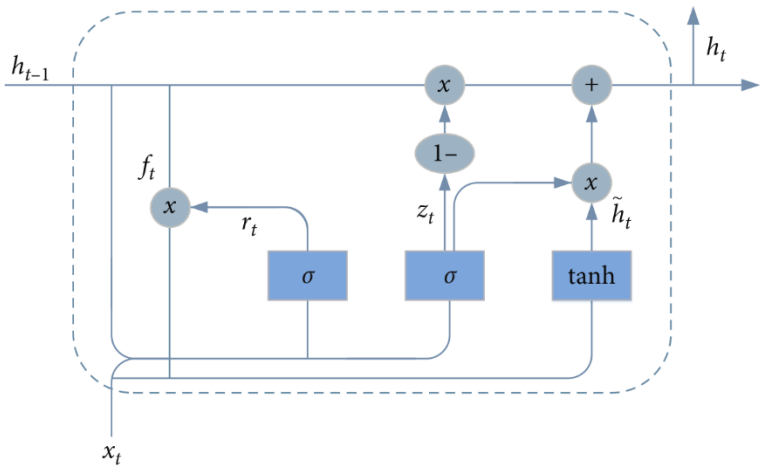
gdje je W_o matrica težina izlaznih vrata i b_o je pomjeraj izlaznih vrata.

Konačno skriveno stanje h_t, koje predstavlja izlaz na svakom vremenskom koraku, se ažurira na osnovu izlaznih vrata i ažuriranog ćelijskog stanja.

$$h_t = o_t \cdot \tanh(C_t). \tag{3.27}$$

3.2.2. GRU

GRU je pojednostavljena verzija LSTM-a koja kombinuje vrata zaboravljanja i ulazna vrata u vrata ažuriranja (eng. *update gate*) i dodaje resetujuća vrata (eng. *reset gate*), te eliminiše ćelijsko stanje, zadržavajući samo skriveno stanje h_t. Na slici 3.6 je dat šematski prikaz arhitekture GRU.



Slika 3.6: Šematski prikaz arhitekture GRU [17]

Ažurirajuća vrata z_t odlučuju koliko od prethodnog skrivenog stanja treba zadržati:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \tag{3.28}$$

gdje je W_z matrica težina ažurirajućih vrata i b_z je pomjeraj ažurirajućih vrata. Dok resetujuća vrata r_t odlučuju koliko informacija iz prethodnog skrivenog stanja treba zaboraviti:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \tag{3.29}$$

gdje je W_r matrica težine reset vrata i b_r je pomjeraj resetujućih vrata.

Generiše se kandidat $\widetilde{h_t}$ za novo skriveno stanje koristeći modifikovano prethodno skriveno stanje:

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h), \tag{3.30}$$

gdje je W_h matrica težina za generisanje kandidata skrivenog stanja i b_h je pomjeraj za kandidat skrivenog stanja. Novo skriveno stanje h_t je linearna kombinacija između prethodnog skrivenog stanja h_{t-1} i kandidata $\widetilde{h_t}$, nivelisana ažurirajućim vratima z_t:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \widetilde{h_t}, \tag{3.31}$$

ako je z_t blizu 0, skriveno stanje će biti približno jednako prethodnom skrivenom stanju h_{t-1}, a ako je z_t blizu 1, skriveno stanje će biti približno jednako kandidatskom stanju $\widetilde{h_t}$.

4. TRANSFORMATORI

RNN pokazuju slabije performanse pri obradi veoma dugih sekvenci, što je posljedica prethodno pomenutih problema eksplozije ili nestajanja gradijenata tokom treniranja. Iako modeli poput LSTM i GRU djelimično ublažavaju ove poteškoće i omogućavaju "pamćenje" informacija kroz duže intervale, njihova sekvencijalna priroda obrade ostaje ključna prepreka. Sekvencijalna obrada onemogućava paralelizaciju tokom treniranja, što usporava proces učenja [18]. Pored toga, uprkos poboljšanjima, i dalje je izazov da informacije s početka sekvence u značajnoj mjeri utiču na elemente koji dolaze kasnije.

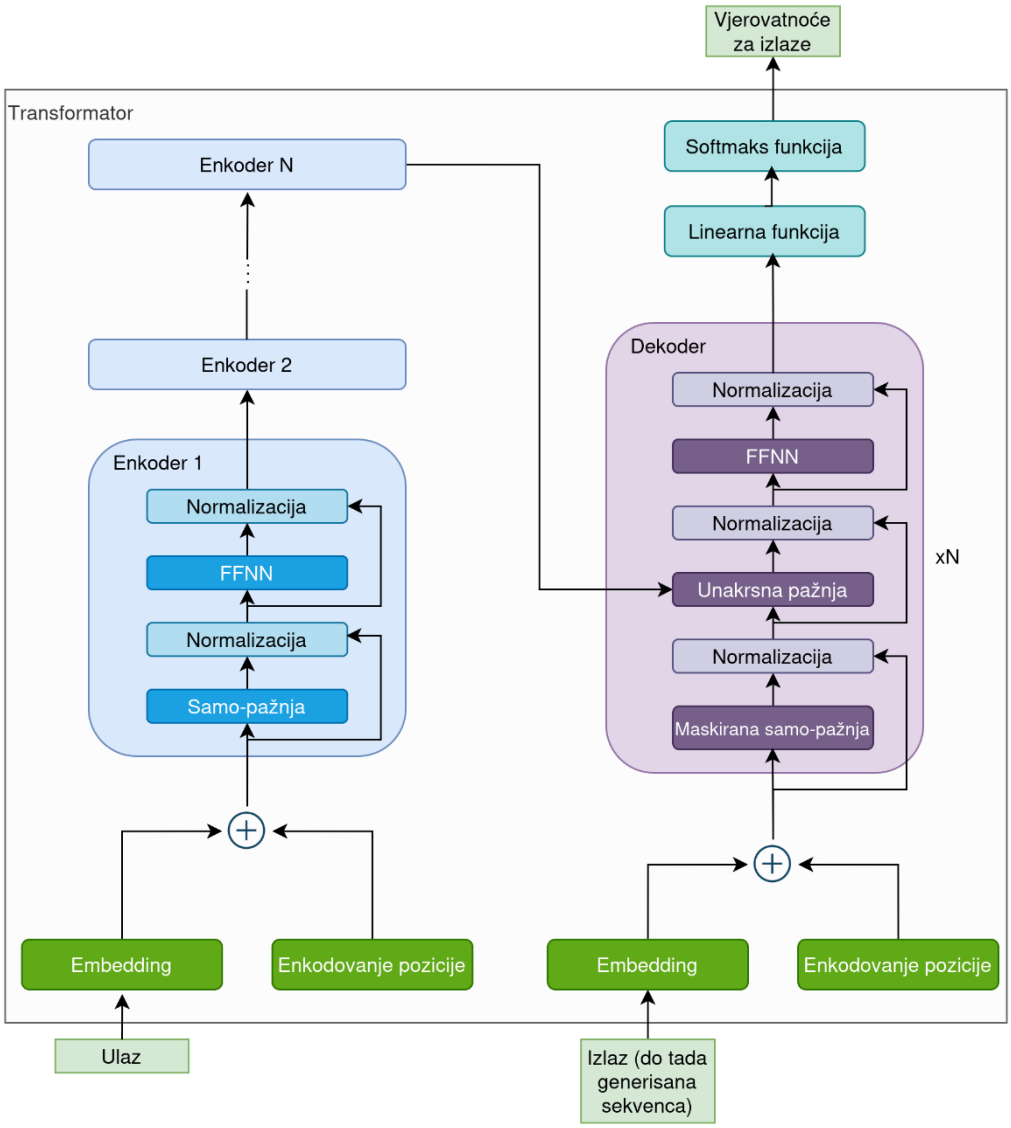
S druge strane, CNN nude prednost paralelne obrade, međutim veličina konvolucionih prozora određuje domet informacija koje model može obraditi u jednom sloju. Ovo dovodi do toga da je za veoma udaljene elemente, pogotovo pri obradi dužih sekvenci, teže uspostaviti jaku međusobnu povezanost [19].

Ova ograničenja motivisala su razvoj nove arhitekture nazvane transformatori, koja se oslanja na tzv. mehanizam pažnje (eng. *attention mechanism*). Mada je mehanizam pažnje ranije bio primjenjivan u RNN modelima, ključna inovacija transformatora jeste upravo iskorišćavanje pažnje kao glavnog gradivnog bloka za obradu sekvenci, bez oslanjanja na rekurentne ili konvolucione slojeve, što je dramatično olakšalo paralelizaciju [19-20]. Mehanizam pažnje omogućava modelu da dinamički određuje na koje dijelove ulazne sekvence treba obratiti pažnju prilikom generisanja svakog izlaznog elementa. Ovo omogućava modelu da direktno poveže svaku riječ u sekvenci sa svim ostalim riječima, bez obzira na njihovu udaljenost, čime se efikasno hvataju dugoročne zavisnosti i odnosi u tekstu.

U radu u kojem su transformatori prvi put predstavljeni, njihova osnovna primjena bila je rješavanje problema prevođenja s jednog jezika na drugi [20]. Međutim, ubrzo je prepoznata univerzalnost i fleksibilnost ove arhitekture, što je dovelo do njene primjene na širok spektar zadataka. Jedan od tih zadataka je i generisanje opisa slika pomoću teksta, gdje su transformatori pokazali bolje rezultate od dotadašnjih pristupa baziranih na kombinaciji CNN+RNN (LSTM ili GRU) [21]. Vremenom su razvijeni i napredniji modeli, poznati kao veliki jezički modeli (eng. *large language models* - LLM), koji su u početku radili isključivo s tekstualnim podacima, prihvatajući tekstualni ulaz i generišući tekstualni izlaz. Kasnije su se pojavile multimodalne varijante ovih modela, sposobne za obradu različitih vrsta multimedijalnog sadržaja, uključujući slike, video i zvuk, pri čemu su se posebno istakle u zadacima poput opisivanja slika, pružajući precizne i prirodne rezultate.

4.1. Arhitektura transformatora

Kao što je prethodno pomenuto, transformatori su prvobitno osmišljeni za rješavanje problema prevođenja, a tek su kasnije uopšteni za rješavanje drugih vrsta problema. Zbog toga je u nastavku prvo dato objašnjenje arhitekture prvobitnog transformatora, nakon toga je opisan transformator koji se koristi za generisanje opisa slika pomoću teksta. Na slici 4.1 je prikazana pojednostavljenja arhitekturalna šema transformatora.



Slika 4.1: Šematski prikaz arhitekture transformatora [22]

Transformatora sastoji se od dva glavna dijela [22]:

1. Enkoder, koji prima ulaznu sekvencu i kreira reprezentacije koje sadrže informacije o cijeloj sekvenci.
2. Dekoder, koji prima reprezentacije iz enkodera i generiše izlaznu sekvencu, element po element, pri čemu koristi informacije iz enkodera i prethodno generisanih izlaza.

Enkoder se sastoji od *N* identičnih slojeva, gdje je svaki sloj izgrađen od dvije komponente:

- Mehanizam samopažnje (eng. *self-attention mechanism*) omogućava da svaka pozicija u enkoderu uzima u obzir sve druge pozicije iz prethodnog sloja. Drugim riječima, svaka riječ u ulaznoj sekvenci može da "obrati pažnju" na sve ostale riječi i uzme u obzir njihov značaj prilikom obrade. Na taj način, model može da prepozna i razumije odnose između bilo koja dva elementa u sekvenci, bez obzira na njihovu udaljenost, što omogućava efikasno hvatanje konteksta i međusobnih uticaja svih riječi u rečenici.

- FFNN-a, poslije primjene mehanizma samopažnje, ulaz se prosljeđuje kroz FFNN. Ova mreža omogućava nelinearne transformacije podataka, što znači da model može da prepozna složenije obrasce i odnose u ulaznim podacima. Ovaj korak dodatno povećava izražajnu moć modela, jer omogućava da model bolje razumije i reprezentuje složene odnose i strukture u podacima.

Dekoder takođe ima N identičnih slojeva, ali svaki sloj sadrži tri ključne komponente:

1. Maskirani mehanizam samopažnje (eng. *masked self-attention mechanism*), funkcioniše slično kao mehanizam samopažnje u enkoderu, razlika je što se koristi maskiranje. Maskiranje sprječava model da "vidi" buduće pozicije tokom predviđanja trenutne pozicije u sekvenci. Npr., ako model generiše rečenicu riječ po riječ, on ne smije da koristi informacije o riječima koje dolaze nakon trenutne riječi, jer bi to značilo da "vara". Umjesto toga, model smije da koristi samo riječi koje su već generisane tj. prethodne riječi.
2. Mehanizam pažnje prema izlazu enkodera, koji se još naziva i mehanizam unakrsne pažnje (eng. *cross-attention mechanism*), koji omogućava dekoderu da obrati pažnju na izlaze enkodera. Drugim riječima, dekoder koristi informacije koje je enkoder već obradio iz ulazne sekvence kako bi generisao odgovarajuću izlaznu sekvencu.
3. FFNN, kao i u enkoderu, ovaj sloj uvodi nelinearnost u model i povećava njegov kapacitet za učenje složenih obrazaca i odnosa.

Ulaz u enkoder transformatora čini tekstualna sekvenca podijeljena na niz tokena iz vokabulara, gdje se svaki token na početku predstavlja vektorskom reprezentacijom (eng. *embedding*) dimenzije d, pri čemu je ta reprezentacija inicijalno slučajna ili unaprijed izračunata, a zatim se koriguje tokom treniranja. Formalno, neka je dat vokabular $V = \{t_1, t_2, \dots, t_m\}$, a funkcija učenja u obliku matrice $E \in \mathbb{R}^{m \times d}$ koja preslikava svaki token t_i u njegov odgovarajući vektorski prikaz $e_i \in \mathbb{R}^d$, tako da, ako je $s = (t_{(1)}, t_{(2)}, \dots, t_{(n)})$ ulazna sekvenca sastavljena od n tokena, odgovarajući vektorski prikazi čine niz $(x_{(1)}, x_{(2)}, \dots, x_{(n)}) = (E[t_{(1)}], E[t_{(2)}], \dots, E[t_{(n)}]) \in \mathbb{R}^{n \times d}$.

Međutim, za razliku od RNN pristupa, gdje je redoslijed ugrađen kroz rekurzivne veze koje prirodno procesuiraju sekvencu korak po korak, transformator obrađuje sve tokene paralelno i stoga eksplicitno uvodi informaciju o poziciji svakog tokena putem pozicioniranja (npr. sinusoidnog kodiranja) p_i, koje se najčešće sabira sa vektorskim prikazom tokena, $(x_{(i)} + p_i)$, kako bi se modelu omogućilo da razlikuje pozicije riječi u rečenici. Ova poziciona informacija neophodna je koliko zbog reprezentacije redoslijeda, toliko i zbog ostajanja dosljednim semantičkim vezama između susjednih i dalekih tokena, budući da sam mehanizam pažnje u transformatoru, definisan skalarnim proizvodom nema ugrađeno razumijevanje sekvencijalnosti pa samim tim treba eksplicitnu pomoć u vidu pozicijskih vektora.

Nakon što je za svaki token dodan pozicijski vektor, rezultujući skup vektora

$$(z_{(1)}, z_{(2)}, \dots, z_{(n)}) = (x_{(1)} + p_1, x_{(2)} + p_2, \dots, x_{(n)} + p_n), \tag{4.1}$$

čini početnu ulaznu reprezentaciju za mehanizam pažnje. U sklopu transformatora koristi se samopažnja sa više glava (eng. *multi-head self-attention*), gdje se za svaki vektor z_i istovremeno računa njegov upit, ključ i vrijednost, te se zatim primjenjuje skalirani proizvod pažnje. Formalno, neka su date linearne projekcije za sve z_i, pri čemu [20, 23]:

- matrica $W^Q \in \mathbb{R}^{d \times d_k}$ preslikava svaku komponentu z_i u prostor upita Q,
- matrica $W^K \in \mathbb{R}^{d \times d_k}$ preslikava svaku komponentu z_i u prostor ključeva K,
- matrica $W^V \in \mathbb{R}^{d \times d_v}$ preslikava svaku komponentu z_i u prostor vrijednosti V.

U matričnom obliku, ako $Z \in \mathbb{R}^{n \times d}$ predstavlja raspored svih z_i redom:

$$Q = ZW^Q, \quad K = ZW^K, \quad V = ZW^V, \tag{4.2}$$

gdje su tada $Q, K \in \mathbb{R}^{n \times d_k}$, a $V \in \mathbb{R}^{n \times d_v}$.

Uobičajeni mehanizam pažnje se može zapisati kao [20]:

$$\text{Pažnja}(Q, K, V) = \text{softmaks}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{4.3}$$

pri čemu se operacija softmaks primjenjuje po svakom redu (odgovaraju za pojedinačni „upit“) i time se dobijaju težinski koeficijenti za kombinovanje vrijednosti (V).

Kod samopažnje sa više glava, koristi se h različitih glava, gdje i-ta glava ima svoje parametre W_i^Q, W_i^K, W_i^V i računa se kao [20, 23]:

$$\text{glava}_i = \text{Pažnja}(Q_i, K_i, V_i) = \text{softmaks}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right)V_i, \tag{4.4}$$

gdje je:

$$Q_i = ZW_i^Q, \quad K_i = ZW_i^K, \quad V_i = ZW_i^V. \tag{4.5}$$

Zatim se sve glava_i konkateniraju i linearno transformišu pomoću matrice $W^O \in \mathbb{R}^{(h \cdot d_v) \times d}$, kako bi se rezultat vratio u prostor dimenzije d, što odgovara dimenziji osnovnih vektorskih reprezentacija u modelu:

$$\text{SamopažnjaSaViše Glava}(Q, K, V) = [\text{glava}_1, \text{glava}_2, \ldots, \text{glava}_h]W^O. \tag{4.6}$$

Izlaz samopažnje sa više glava se prosljeđuje kroz dodatne slojeve poput sloja normalizacije, te FFNN unutar svakog bloka transformatora.

Dekoder tokom generisanja izlazne sekvence mora “korak po korak” da predviđa sljedeći token, pri čemu se u svakom koraku koriste svi do tada određeni izlazni tokeni, dok su budući tokeni sakriveni pomoću maskiranja [23, 24]. Suštinski se na izlaz samopažnje dodaje matrica M koja ima vrijednost $-\infty$ za buduće elemente i vrijednost 0 za već viđene elemente. Pošto na početku nema prethodno određenih tokena, koristi poseban token koji označava početak sekvence. Izlazi iz maskirane samopažnje se koriste u narednom sloju za kreiranje upita, dok se ključevi i vrijednosti dobijaju na osnovu izlaza posljednjeg enkodera, koji se prosljeđuje svim dekoderima istovremeno. Formalno neka su:

- $Z^D \in \mathbb{R}^{n' \times d}$: reprezentacija svih skrivenih vektora dekodera,
- $Z^E \in \mathbb{R}^{n \times d}$: konačni izlaz enkodera.

Svaka komponenta Z^D (svaki z_i^D) projektuje se u prostor upita, a ključevi i vrijednosti dolaze iz linearnog mapiranja Z^E. Za to se koriste tri matrice:

- $W_D^Q \in \mathbb{R}^{d \times d_k}$, preslikava Z^D u prostor upita Q^D,
- $W_D^K \in \mathbb{R}^{d \times d_k}$, preslikava Z^E u prostor ključeva K^E,
- $W_D^V \in \mathbb{R}^{d \times d_v}$:, preslikava Z^E u prostor vrijednosti V^E.

Rezultat unakrsne samopažnje je dat izrazom:

$$\text{Pažnja}(Q^D, K^E, V^E) = \text{softmaks}\left(\frac{Q^D (K^E)^\top}{\sqrt{d_k}}\right) V^E. \tag{4.7}$$

Kao i kod samopažnje, unakrsna pažnja implementira varijantu sa više glava. Parametri (W_i^Q, W_i^K, W_i^V) definišu se za svaku glavu, i kreiraju se projekcione matrice:

$$Q_i^D = Z^D W_i^Q, \quad K_i^E = Z^E W_i^K, \quad V_i^E = Z^E W_i^V. \tag{4.8}$$

Pa se za svaku glavu računa:

$$\text{glava}_i = \text{softmaks}\left(\frac{Q_i^D (K_i^E)^\top}{\sqrt{d_k}}\right) V_i^E. \tag{4.9}$$

Sve glave se konkateniraju i linearno transformišu pomoću matrice $W_D^O \in \mathbb{R}^{(h \cdot d_v) \times d}$:

$$\text{SamopažnjaSaVišeGlava}(Q^D, K^E, V^E) = [\text{glava}_1, \text{glava}_2, \ldots, \text{glava}_h] W^O. \tag{4.10}$$

Nakon N blokova maskirane samopažnje, unakrsne pažnje i FFNN-a, formira se niz skrivenih vektora dimenzije d, označen kao $Z_N^D \in \mathbb{R}^{n' \times d}$, pri čemu svaki vektor odgovara jednoj poziciji u izlaznoj sekvenci. Ti se vektori linearno mapiraju u logite, odnosno ocjene za sve riječ iz vokabulara. Npr., ako konačna izlazna rečenica ima sedam tokena, a vokabular ciljnog jezika 10.000 riječi, model generiše 7×10.000 logita. Logiti se dovode na softmaks sloj. Kod prvobitne verzije transformatora se birao najvjerovatniji token prema izlazu softmaks funkcije, a kod kasnijih transformatora se koristi stohastičko uzrokovanje, takođe kod novijih transformatora se uvodi modifikacija u vidu hiperparametra koji se naziva temperatura. Za svaki logit $L_{i,j}$, vjerovatnoća $P_{i,j}$ se računa prema formuli:

$$P_{i,j} = \frac{e^{L_{i,j}/T}}{\sum_{k=1}^{V} e^{L_{i,k}/T}}. \tag{4.11}$$

Kada je temperatura manja od jedan, najveće vrijednosti logita će postati još dominantnije, a vjerovatnoće za manje vjerovatne riječi će se dodatno smanjiti. Rezultat je da model postaje deterministčniji, jer će birati najvjerovatniji token s gotovo potpunom sigurnošću. Kada je temperatura veća od jedan, vrijednosti logita se "izravnavaju", pa razlike između njih postaju manje izražene. Kao rezultat, vjerovatnoće za manje vjerovatne riječi postaju veće, a

distribucija vjerovatnoća postaje ravnomjernija. Ovo omogućava modelu da bira i manje vjerovatne riječi, što povećava raznolikost i kreativnost u generisanom tekstu [24].

Kada je temperatura jednaka jedan, dobija se standardni izraz za softmaks funkciju. U ovom slučaju, vjerovatnoće direktno odražavaju razlike između logita. Treniranje transformatora se vrši pomoću prethodno opisane propagacije unazad.

4.2. Arhitektura transformatora za generisanje opisa slika

Arhitektura transformatora za zadatak opisivanja slika se sastoji od dva glavna modula: vizuelnog enkodera, koji ekstrahuje reprezentacije iz ulazne slike, i tekstualnog dekodera, koji na osnovu tih reprezentacija generiše opis u prirodnom jeziku [21].

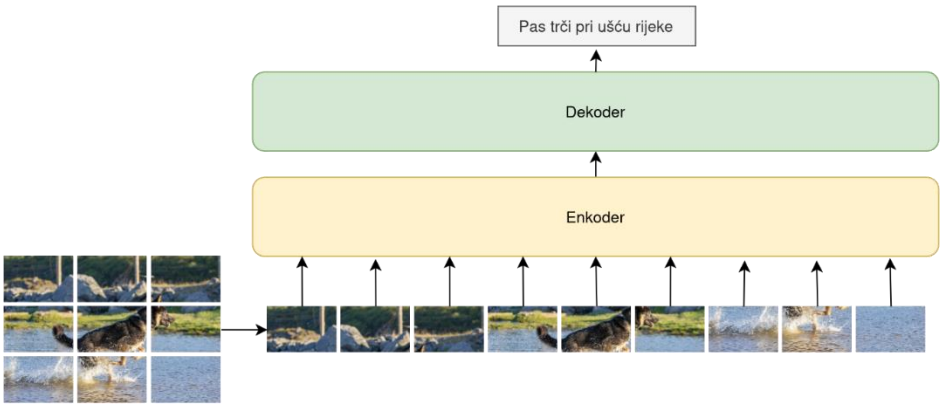
Vizuelni enkoder koristi unaprijed obučene modele poput *Vision Transformer* (ViT) ili SWIN transformatora kako bi se iz ulazne slike generisale bogate reprezentacije koje sažimaju njene semantičke i prostorne karakteristike. Formalno, neka je ulazna slika predstavljena kao matrica piksela $\boldsymbol{I} \in \mathbb{Z}^{H \times W \times C}$, gdje su H, W, i C visina, širina i broj kanala slike, respektivno. Kako bi se slika mogla obraditi kroz transformator, neophodno je izvršiti njenu transformaciju u niz vektorskih reprezentacija.

Prvi korak u obradi slike je njena podjela na isječke (eng. *patches*) dimenzija $P \times P$. Svaki blok se zatim transformiše u vektor fiksne dimenzije pomoću linearne projekcije. Ako je broj zakrpa po slici $N = \frac{H}{P} \cdot \frac{W}{P}$, tada se svaka zakrpa $\boldsymbol{p_i} \in \mathbb{Z}^{P \times P \times C}$ preslikava u vektor $\boldsymbol{z_i} \in \mathbb{R}^d$ pomoću matrice projekcije $\boldsymbol{W_p} \in \mathbb{R}^{(P \cdot P \cdot C) \times d}$.

Kao i kod tekstualnog transformatora, da bi model mogao da razlikuje prostorne odnose između isječaka, svakom vektoru $\boldsymbol{z_i}$ dodaje se odgovarajući pozicioni vektor $\boldsymbol{p_i} \in \mathbb{R}^d$. Na taj način se dobija ulazna reprezentacija za transformator:

$$\boldsymbol{z_i^{(0)}} = \boldsymbol{z_i} + \boldsymbol{p_i}, \quad i = 1, \ldots, N. \tag{4.12}$$

Vizuelni enkoder takođe koristi samopažnju sa više glava. Nakon L slojeva samopažnje sa više glava i FFNN-a, vizuelni enkoder generiše niz vektora $\boldsymbol{Z_E} \in \mathbb{R}^{N \times d}$, koji predstavljaju sažete reprezentacije ulazne slike. Ovi vektori služe kao ulaz za tekstualni dekoder. Dekoder je klasičan tekstualni dekoder koji je prethodno opisan i sadrži sve iste komponente. Na slici 4.2 je prikazana pojednostavljena arhitekturalna šema datog transformatora.



Slika 4.2: Šematski prikaz pojednostavljene arhitekture transformatora za opisivanje slika

Postoje različite implementacije date arhitekture transformatora za generisanje tekstualnih opisa slika kao što su GIT, Florence2, BLIP2, koje su korištene za rješavanje nekih od problema opisanih u narednim poglavljima i čija je evaluacija data u praktičnom dijelu rada.

4.3. Veliki jezički modeli

Iako se transformatori tradicionalno sastoje od enkodera i dekodera, s vremenom su razvijene varijante koje koriste samo enkoder i služe isključivo za zadatke klasifikacije. Jedan od najpoznatijih primjera takve arhitekture je BERT, koji će biti detaljnije opisan u narednom poglavlju. S druge strane, pojavile su se i varijante koje koriste isključivo dekoder, a njihova glavna primjena je predikcija narednog tokena. Ova arhitektura je osnova velikih jezičkih modela kao što su GPT, Llama, Claude i drugi.

Kod modela sa enkoderom, ulazna sekvenca se transformiše u sažetu reprezentaciju koja sadrži ključne informacije. Nasuprot tome, transformatori zasnovani samo na dekoderu predstavljaju čitavu ulaznu sekvencu kroz odgovarajuće vektorske reprezentacije, bez sažimanja. Pri generisanju svakog narednog tokena, model koristi i originalni ulaz i sve prethodno generisane tokene, čime se omogućava progresivno generisanje teksta.

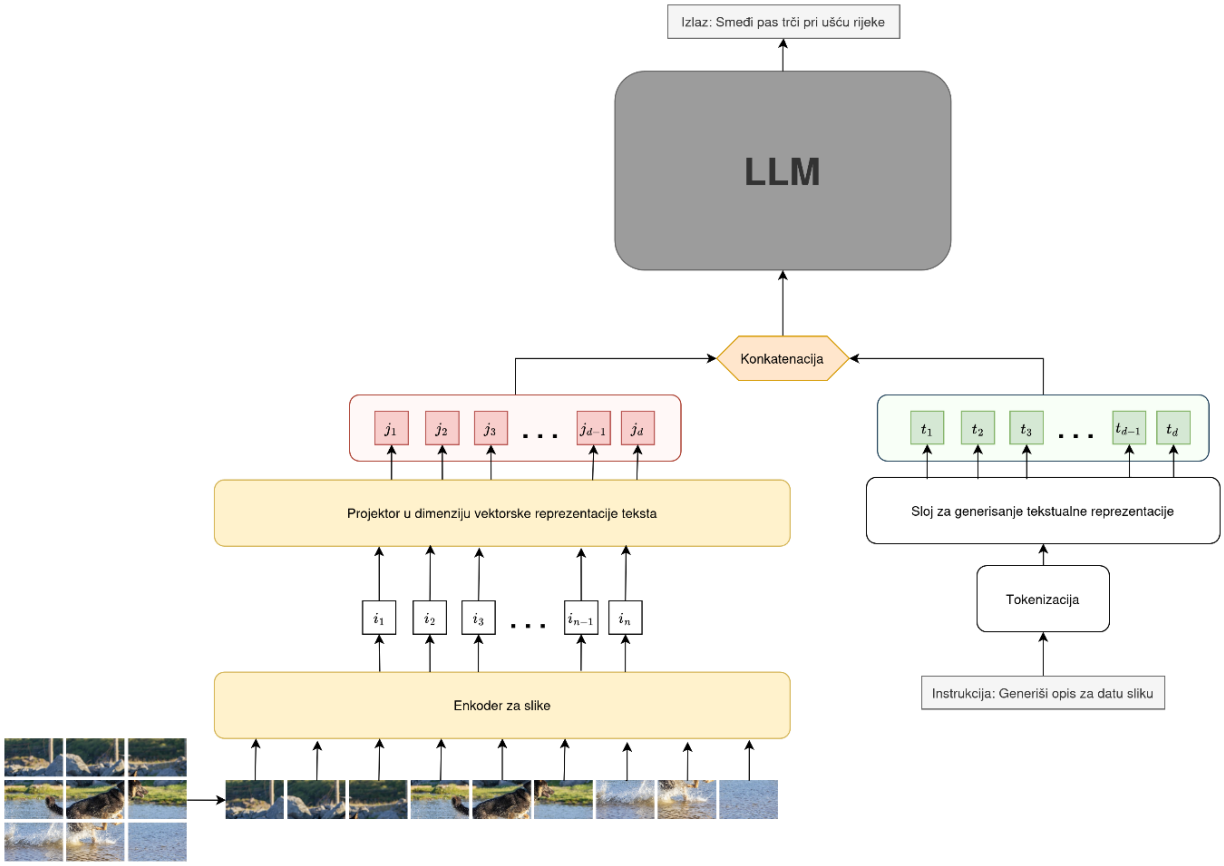
LLM-ovi predstavljaju pravu revoluciju u svijetu vještačke inteligencije, jer omogućavaju rješavanje širokog spektra problema na način koji ranije nije bio moguć. Dok su stariji modeli bili ograničeni na rješavanje specifičnih zadataka ili usko definisanih skupova problema, LLM-ovi su prvi univerzalno primjenjivi modeli [25]. Za LLM-ove je karakteristično da se kao ulaz prihvata tokenizovana vektorska reprezentacija teksta koji predstavlja instrukciju za LLM pomoću koje se on koristi da rješava neki specifičan zadatak, za koji nije direktno obučen. Ova sposobnost da model generalizuje i rješava zadatke bez prethodnog eksplicitnog nadgledanog treniranja naziva se *zero-shot learning*[5]. Pokazalo se da kvalitet generisanog odgovora uveliko zavisi od kvaliteta napisane instrukcije koja je data na ulazu LLM-a, a postupak kojim se oblikuje instrukcija se naziva *prompt engineering* [25]. O kome će biti više riječi u praktičnom dijelu rada.

Prvobitni LLM-ovi su radili isključivo s tekstualnim ulazima, što je značilo da su njihove mogućnosti bile ograničene. Tek razvojem multimodalnih LLM-ova, koji mogu obrađivati i slike, audio snimke i druge vrste multimedijalnog sadržaja, otvoren je put ka modelima sposobnim za dublje razumijevanje i kontekstualnu integraciju informacija iz različitih izvora. Postoje dva glavna pristupa za implementaciju multimodalnih LLM-ova [26]:

- Model koristi jedinstveni dekoder za obradu svih modaliteta, koristi se od strane modela poput GPT i Llama i detaljno je detaljnije opisan u nastavku, i
- model koristi unakrsnu pažnju između različitih modaliteta.

Na slici 4.3 je prikazana pojednostavljena šematska arhitektura modela koji koristi jedinstven dekoder za obradu svih modaliteta, u ovom slučaju tekstualnog i vizuelnog, tj. rasterske slike.

[5] Pored *zero-shot* pristupa, postoje i *one-shot* i *few-shot* pristupi, gdje model dobija jedan ili nekoliko primjera rješenja zadatka uz instrukciju, čime se poboljšava tačnost odgovora i omogućava bolje prilagođavanje zadatku.



Slika 4.3: Šematski prikaz pojednostavljene arhitekture multimodalnog LLM

Tekst koji ulazi u dekoder najprije se tokenizuje i predstavlja pomoću odgovarajuće vektorske reprezentacije za tekst. Analogno tome, na strani slike vrši se enkodovanje slike pomoću vizuelnog transformatora, zatim se vektor $\boldsymbol{i} \in \mathbb{R}^n$, dobijen na izlazu enkodera, projektuje u prostor dimenzije d pomoću funkcije $P: \mathbb{R}^n \to \mathbb{R}^d$. Na taj način se dobije vektor $\boldsymbol{j} \in \mathbb{R}^d$. U sljedećoj fazi, dobijeni vektor $\boldsymbol{j}$ se spaja sa vektorom $\boldsymbol{t} \in \mathbb{R}^d$, koji predstavlja vektorsku reprezentaciju tokenizovanog tekstualnog ulaza tj. instrukcije, čime se formira zajednički ulaz za LLM $[j_1, j_2, \dots, j_d, t_1, t_2, \dots, t_d]$. Koji potom prolazi kroz LLM:

$$\boldsymbol{t}_{\text{izlaz}} = \text{LLM}([j_1, j_2, \dots, j_d, t_1, t_2, \dots, t_d]), \tag{4.13}$$

pri čemu $\boldsymbol{t}_{\text{izlaz}}$ predstavlja vektorsku reprezentaciju izlaznih tokena. Nakon dekodovanja, se dobija tekstualni opis slike.

5. OBRADA PRIRODNOG JEZIKA

Obrada prirodnog jezika (eng. *natural language processing* - NLP) je skup metoda koje imaju za cilj da ljudski (prirodni) jezik učine razumljivim računarima. NLP omogućava pretraživanje teksta, ekstrakciju i razumijevanje informacija iz teksta, prevođenje između jezika, analizu sentimenta, generisanje tekstualnog opisa slika na osnovu prepoznatih objekata i scena, kao i razne druge stvari. Za rješavanje navedenih problema NLP se oslanja na ML tehnike [27].

Tekst je vremenski nezavisan sadržaj sa linearnom strukturom, a u računarima je kodovan u binarnom obliku [10]. Kako bi se tekstualni sadržaj tumačio pomoću NLP metoda, neophodno je kreirati odgovarajuću vektorsku reprezentaciju. Što je reprezentacija bogatija informacijama, to se efikasnije može primijeniti u različitim NLP zadacima. Reprezentacije variraju od jednostavnih, kao što je *one-hot encoding*, do onih složenijih, o kojima je više dato u nastavku.

U nastavku su opisane i metrike koje se koriste za evaluaciju kvaliteta generisanih opisa za slike.

5.1. Reprezentacije bazirane na frekvenciji riječi

Reprezentacije bazirane na frekvenciji riječi koriste informacije o učestalosti svake riječi u dokumentu kako bi oblikovale numerički prikaz teksta. U ovim modelima, tekst se posmatra kao skup riječi gdje se vrijednosti u vektorima zasnivaju na frekvenciji ili prisutnosti pojedinačnih riječi. Takvi modeli zanemaruju redoslijed riječi i sintaksičke veze, ali pružaju jednostavan način za analizu sadržaja teksta na osnovu zastupljenosti riječi.

5.1.1. Bag of Words

Bag of Words (BoW) je jednostavan i intuitivan model za predstavljanje teksta u numeričkom obliku. Ovaj model posmatra tekst (kao što je rečenica ili dokument) kao vreću riječi, zanemarujući gramatiku, redoslijed riječi i sintaksičke veze između njih. Fokus je isključivo na prisustvu i frekvenciji riječi unutar teksta. Formiranje BoW modela podrazumijeva kreiranje vokabulara (skup jedinstvenih riječi) iz skupa dokumenata, a zatim predstavljanje svakog dokumenta kao vektora sa dimenzijama jednakim broju riječi u vokabularu. Vrijednosti u vektoru mogu biti jednostavna frekvencija riječi ili binarne vrijednosti koje označavaju prisustvo/odsustvo riječi. Formalno, neka je $D = \{d_1, d_2, \dots, d_n\}$ skup dokumenata, i neka je $V = \{w_1, w_2, \dots, w_m\}$ vokabular sa m riječi. Svaki dokument d_i se reprezentuje kao vektor $\overrightarrow{v_i} = (c_1, c_2, \dots, c_m)$, gdje je c_j broj pojavljivanja riječi w_j u dokumentu d_i.

Primjer 5.1: Neka su data četiri dokumenta:

- d_1: "Mačka lovi miševe.",
- d_2: "Pas laje noću.",
- d_3: "Miš se krije od mačke.",
- d_4: "Od se krije miš mačke".

Vokabular V $= \{\text{mačka}, lovi, miševe, pas, laje, noću, miš, se, krije, od, mačke\}$.

BoW reprezentacija:

- d_1: [1,1,1,0,0,0,0,0,0,0,0],
- d_2: [0,0,0,1,1,1,0,0,0,0,0],
- d_3: [0,0,0,0,0,0,1,1,1,1,1],
- d_4: [0,0,0,0,0,0,1,1,1,1,1].

Zbog navedenih ograničenja BoW reprezentacija za dokumente d_3 i d_4 je jednaka, iako d_4 nema logički smisao kad se interpretira kao prirodni jezik. BoW model ne razlikuje informativne od uobičajenih riječi (poput „i“, „je“, „od“), koje se često pojavljuju u većini dokumenata i tako smanjuju važnost relevantnih riječi. Takođe, kada vokabular sadrži veliki broj riječi, vektori postaju visokodimenzionalni, što otežava obradu i smanjuje efikasnost, posebno u složenijim NLP zadacima.

5.1.2. TF-IDF

Da bi se riješio problem neinformativnih riječi i izdvojile relevantne riječi, koristi se *Term Frequency-Inverse Document Frequency* (TF-IDF) reprezentacija. TF-IDF kombinuje frekvenciju pojavljivanja riječi u dokumentu $TF_{t,d}$ sa mjerom rijetkosti (eng. *inverse document frequency*) te riječi u kolekciji IDF_t, kako bi se smanjila težina riječi koje su česte u svim dokumentima, a povećala težina specifičnih riječi. Formalno:

$$IDF_t = \log\left(\frac{N}{DF_t}\right), \tag{5.1}$$

gdje je N ukupan broj dokumenata, a DF_t broj dokumenata koji sadrže riječ t. Ako se riječ pojavljuje u mnogim dokumentima, DF_t je velik, pa IDF_t postaje mali, što smanjuje njenu težinu. TF-IDF je tada definisan kao:

$$TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t. \tag{5.2}$$

Primjer 5.2: Neka je data kolekcija sa 500.000 dokumenata. U tabeli 5.1 su prikazane frekvencija termina za tri dokumenta, a inverzne frekvencije su date u tabeli 5.2. Na osnovu njih izračunate su TF-IDF vrijednosti koje su date u tabeli 5.3.

Tabela 5.1: Frekvencije pojavljivanja termina

Termin	Dok1	Dok2
program	15	5
mreža	2	25
podaci	0	20
sistem	10	0

Tabela 5.2: Inverzne frekvencije dokumenata u kolekciji sa 500.000 dokumenata

Termin	DF_1	IDF_1

program	25000	1,8
mreža	12000	2,2
podaci	22000	1,9
sistem	30000	1,65

Tabela 5.3: Izračunate TF-IDF težine

Termin	**Dok1**	**Dok2**	**Dok3**
program	27	9	21,6
mreža	4,4	55	6,6
podaci	0	38	34,2
sistem	16,5	0	24,75

Vektori dobijeni pomoću TF-IDF reprezentacije su rijetko popunjeni, pošto se kao kod BoW u svakom dokumentu, uglavnom pojavljuje samo mali broj od svih riječi iz vokabulara. Iako TF-IDF omogućava izdvajanje relevantnih riječi, i dalje ignoriše kontekst u kojem se riječi pojavljuju i ne može da razlikuje značenje sinonima ili polisemiju (riječ sa više značenja). Reprezentacije zasnovane na frekvenciji, kao što su BoW i TF-IDF, predstavljaju tekst kao skup izolovanih riječi bez razumijevanja njihovih međusobnih odnosa. Da bi se prevazišla neka od ovih ograničenja, razvijene su reprezentacije zasnovane na ugrađivanju riječi (eng. *word embeddings*).

5.2. Reprezentacije bazirane na ugrađivanju riječi

Reprezentacije bazirane na ugrađivanju riječi omogućavaju kodiranje semantičkih i sintaksičkih odnosa između riječi. Ugrađivanje riječi je tehnika koja svakom pojmu u jeziku pridružuje gusto popunjeni vektor niske dimenzionalnosti, čime se omogućava zadržavanje informacija o sličnosti između riječi, kao i odnosa među njima. Cilj ovih metoda je smanjiti dimenzionalnost podataka uz očuvanje ključnih informacija o značenju i kontekstu. Tehnike ugrađivanja riječi bazirane su na principu da se slične riječi, s obzirom na njihovo značenje i kontekst upotrebe, moraju nalaziti blizu jedna druge u prostoru vektorskih reprezentacija. Zajednički nedostatak ovih vektorskih reprezentacije je njihova nemogućnost da riješe problem polisemije, jer dodjeljuju isti vektor višeznačnim riječima bez obzira na kontekst [18].

5.2.1. Word2Vec

Word2Vec je algoritam razvijen od strane istraživača u Google-u[6]. Koristi neuronske mreže za učenje vektorskih reprezentacija riječi na osnovu njihovog konteksta u tekstualnim podacima. Postoje dvije osnovne arhitekture koje koristi Word2Vec za učenje ovih vektora: *Continuous Bag of Words* (CBOW) i *Skip-Gram* (SG) [28].

CBOW model predviđa ciljnu riječ koristeći njene susjedne riječi. Drugim riječima, za svaku riječ w_t, model pokušava predvidjeti w_t na osnovu njenih susjednih riječi maksimizujući

[6] https://www.google.com/

vjerovatnoću $P(w_t|w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n})$. Matematički, funkcija cilja CBOW modela za minimizaciju greške može se izraziti kao [28]:

$$\frac{1}{T}\sum_{t=1}^{T} \log P\left(w_t|w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}\right) = \frac{1}{T}\sum_{t=1}^{T} \log P\left(w_t \mid v_{w_C}\right). \quad (5.3)$$

Nasuprot tome, SG koristi ciljnu riječ da bi predvidio njene susjedne riječi. Dakle, za svaku riječ w_t, model pokušava predvidjeti kontekstualne riječi maksimizujući vjerovatnoću $P(w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}|w_t)$. Matematički, optimizacijska funkcija SG modela za minimizaciju greške može se izraziti kao [28]:

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-n\le j\le n, j\neq 0} \log P\left(w_{t+j}|w_t\right), \quad (5.4)$$

gdje je T broj riječi u korpusu, a n broj kontekstualnih riječi s obje strane ciljne riječi. Raspodjela vjerovatnoće $P\left(w_{t+j}|w_t\right)$ modeluje se korišćenjem softmaks funkcije:

$$P(w_O|w_I) = \frac{\exp\left(v'_{w_O} \cdot v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left(v'_w \cdot v_{w_I}\right)}, \quad (5.5)$$

gdje v_{w_I} predstavlja vektor ulazne riječi, v'_{w_O} vektor izlazne riječi, a W ukupni broj riječi u vokabularu. Ova formulacija je nepraktična jer su troškovi računanja $\nabla \log p\left(w_O|w_I\right)$ proporcionalni sa W, što je često veliko (reda 10^5 do 10^7 članova). Da bi se ubrzalo izračunavanje softmaks funkcije, Word2Vec koristi aproksimacije pomoću negativnog uzorkovanja (eng. *negative sampling*). Umjesto ažuriranja težina za sve riječi u vokabularu, model ažurira samo težine za pozitivne parove (ciljna i kontekstualna riječ) i nekoliko negativnih uzoraka, koji su nasumično odabrani. Funkcija cilja negativnog uzorkovanja je:

$$\log \sigma\left(v'^{\top}_{w_O} v_{w_I}\right) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}\left[\log \sigma\left(-v'^{\top}_{w_i} v_{w_I}\right)\right] \quad (5.6)$$

gdje je k broj negativnih uzoraka, a $P_n(w)$ distribucija negativnih uzoraka, obično proporcionalna frekvenciji riječi podignutoj na neki stepen, obično $3/4$.

Word2Vec se pokazao kao efikasan model za generisanje vektorskih reprezentacija riječi, posebno u radu sa velikim vokabularima koji sadrže milione riječi. Međutim, Word2Vec ima i nekoliko značajnih nedostataka. Prvo, Word2Vec koristi lokalni kontekstni prozor, što znači da zanemaruje globalne korelacije između udaljenih riječi u tekstu, čime se gube važne semantičke informacije. Ovaj problem djelimično rješava GloVe. Drugo značajno ograničenje Word2Vec-a je njegova nemogućnost rada sa nepoznatim riječima, odnosno riječima koje nisu bile prisutne u trening skupu. Ovo je posebno problematično u kontekstu jezika sa bogatom morfologijom, kao što je srpski jezik. Ovaj problem je efikasno riješen sa FastText.

5.2.2. FastText

FastText je model za ugrađivanje riječi razvijen od strane istraživača u Facebook-u[7], koji zadržava osnovne principe Word2Vec-a. Ovaj model uvodi koncept podriječi (eng. *subwords*), što omogućava da se riječi predstavljaju kao skup slovnih n-grama, pružajući fleksibilniju i informativniju vektorsku reprezentaciju. Na taj način, vektorska reprezentacija riječi je dobijena kao zbir vektora njenih n-gramova. Na primjer, riječ "bašta" može biti podijeljena na n-grame dužine 3, kao što su "<ba", "baš", "ašt", "šta" i "ta>". Ovdje su dodani specijalni simboli "<" i ">" kako bi se označio početak i kraj riječi, čime se omogućava modelu da razlikuje prefikse i sufikse od unutrašnjih dijelova riječi. Ovakav pristup omogućava FastText-u da efikasno generalizuje i na riječi koje nisu prisutne u trening skupu, jer se vektorska reprezentacija nepoznate riječi može formirati na osnovu n-gramova koje dijeli sa poznatim riječima [29-30].

Formalno, neka je w riječ iz vokabulara, a $G(w)$ skup slovnih n-grama koji čine tu riječ. FastText model tada predstavlja riječ w kao zbir vektora njenih slovnih n-gramova:

$$v(w) = \sum_{g \in G(w)} v(g), \tag{5.7}$$

gdje je $v(g)$ vektor slovnog n-grama g. Ova reprezentacija omogućava modelu da zadrži morfološke i sintaksne informacije o riječi, što je posebno korisno za jezike sa složenom morfologijom, poput srpskog. Na primjer, različiti oblici riječi "mačka" (kao što su "mačke", "mačkom", "mačkama") dijele mnoge iste n-gramove, što omogućava modelu da prepozna njihovu sličnost i generiše slične vektorske reprezentacije.

FastText koristi istu arhitekturu kao Word2Vec, odnosno podržava i CBOW i SG modele. U CBOW varijanti, model koristi n-gramove podriječi susjednih riječi da bi predvidio ciljnu riječ, dok u SG varijanti model koristi n-gramove podriječi ciljne riječi kako bi predvidio susjedne riječi. FastText takođe koristi negativno uzorkovanje kako bi ubrzao proces treniranja, slično kao Word2Vec. Funkcija cilja koja se koristi u FastText-u sa negativnim uzorkovanjem može se izraziti kao [29]:

$$\log \sigma\left(v(w) \cdot v(c)\right) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}\left[\log \sigma\left(-v(w_i) \cdot v(c)\right)\right], \tag{5.8}$$

gdje je w ciljna riječ, c kontekstualna riječ, k broj negativnih uzoraka, a $P_n(w)$ distribucija negativnih uzoraka.

Jedan od ključnih izazova u radu sa FastText-om je izbor optimalne dužine slovnih n-grama. Prekratki n-grami mogu izgubiti semantičke informacije, dok predugi n-grami mogu dovesti do prekomjernog prilagođavanja modela specifičnim riječima. U praksi, n-grami dužine između tri i šest karaktera pokazali su se kao optimalni za većinu jezika, uključujući srpski.

[7] https://about.meta.com/

5.2.3. GloVe

GloVe je model za ugrađivanje riječi koji su razvili istraživači sa Stanforda[8], a koji predstavlja napredak u odnosu na prethodne modele poput Word2Vec-a. Ključna inovacija GloVe-a leži u njegovom korišćenju matričnih faktorizacija i globalnih informacija o distribuciji riječi, što rezultuje vektorskim reprezentacijama koje reflektuju kako lokalne, tako i globalne odnose u tekstu.

Osnovna ideja GloVe-a je da se statističke informacije o ko-pojavljivanju riječi koriste za izgradnju vektorskih reprezentacija. GloVe model konstruiše matricu ko-pojavljivanja, pri čemu svaki element X_{ij} označava broj pojavljivanja riječi i u kontekstu riječi j unutar korpusa. Na taj način model hvata globalne odnose između riječi [30-31]. Cilj GloVe-a je da faktorizuje ovu matricu i pronađe vektore riječi w_i i w_j koji zadovoljavaju relaciju:

$$w_i^T w_j + b_i + b_j = \log(X_{ij}), \tag{5.9}$$

gdje su w_i i w_j vektori riječi i i j, dok su b_i i b_j skalari koji predstavljaju pomjeraje za svaku riječ. Logaritamska transformacija se koristi kako bi se smanjila velika odstupanja u frekvencijama ko-pojavljivanja, čime se omogućava bolje modelovanje odnosa između riječi koje se pojavljuju rijetko i onih koje se pojavljuju često.

GloVe koristi funkciju greške koja minimizuje razliku između predviđene vrijednosti $w_i^T w_j + b_i + b_j$ i stvarne vrijednosti $\log(X_{ij})$. Ova funkcija greške je ponderisana tako da riječi koje se rijetko pojavljuju zajedno imaju manji uticaj na ukupnu grešku, dok riječi koje se često pojavljuju zajedno imaju veći uticaj. Funkcija greške GloVe modela može se formalno zapisati kao:

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(w_i^T w_j + b_i + b_j - \log(X_{ij})\right)^2, \tag{5.10}$$

gdje je V veličina vokabulara, a $f(X_{ij})$ ponder funkcija koja smanjuje uticaj rijetkih ko-pojavljivanja, dok povećava uticaj čestih ko-pojavljivanja. Funkcija $f(X_{ij})$ je definisana kao:

$$f(X_{ij}) = min\left(\left(X_{ij}/X_max\right)^{\alpha}, 1\right), \tag{5.11}$$

gdje su $X_{\max}$ i α hiperparametri koji kontrolišu oblik ponder funkcije. Ova funkcija omogućava da se model fokusira na relevantne ko-pojave riječi, dok zanemaruje slučajeve gdje su ko-pojave riječi previše rijetke da bi bile statistički značajne.

GloVe koristi cijeli korpus kako bi izgradio svoju matricu ko-pojavljivanja. Ovaj globalni pristup omogućava da se uhvate odnose između riječi koje se možda nikada ne pojavljuju zajedno u istom lokalnom kontekstu, ali koje dijele slične obrasce ko-pojavljivanja sa drugim riječima. Na primjer, riječi poput "kralj" i "kraljica" mogu imati slične obrasce ko-pojavljivanja sa riječima poput "kruna", "prijesto" ili "monarhija", iako se možda ne pojavljuju zajedno u istoj rečenici. GloVe takođe omogućava da se u vektorskom prostoru modeluju linearni odnosi između riječi. Na primjer, poznata je sposobnost GloVe-a da uhvati analogije poput "kralj" :

[8] https://www.stanford.edu/

"kraljica" = "muškarac" : "žena". Ovi linearni odnosi proizlaze iz činjenice da su vektorske reprezentacije riječi u GloVe-u organizovane tako da razlike između vektora reflektuju semantičke razlike između riječi. Na primjer, vektor razlike između riječi "kralj" i "kraljica" može biti sličan vektoru razlike između riječi "muškarac" i "žena", što omogućava modelu da prepozna analogne odnose između različitih parova riječi.

5.3. Reprezentacije bazirane na kontekstualnim jezičkim modelima

Reprezentacije bazirane na kontekstualnim jezičkim modelima uzimaju u obzir kontekst u kojem se riječ pojavljuje, što znači da generišu različite vektorske reprezentacije za istu riječ u različitim kontekstima, čime se efikasno rješava problem polisemije i sinonimije (različite riječi sa sličnim značenjem). Ovi modeli koriste arhitekture zasnovane na dubokim neuronskim mrežama. Posebno se izdvajaju transformatorske arhitekture zbog svoje sposobnosti da analiziraju cijeli tekstualni kontekst. Njihova efikasnost proističe iz treniranja na velikim količinama podataka.

5.3.1. ELMo

ELMo je kontekstualni jezički model razvijen od strane istraživača sa Univerziteta u Vašingtonu[9]. ELMo predstavlja napredak u oblasti ugrađivanja riječi jer uvodi koncept dinamičkih, kontekstualnih reprezentacija. Osnovna arhitektura ELMo modela zasniva se na dvosmjernim LSTM mrežama (eng. *bidirectional* LSTM - BiLSTM). BiLSTM omogućava modelu da obrađuje tekst u oba smjera. Ova dvosmjerna analiza omogućava modelu da uhvati bogatije kontekstualne informacije jer svaka riječ može biti interpretirana u odnosu na riječi koje joj prethode, ali i na riječi koje dolaze poslije nje [32].

ELMo koristi slovne, odnosno znakovne CNN za generisanje inicijalnih vektorskih reprezentacija riječi na nivou znakova, čime se efikasno nosi sa nepoznatim i rijetkim riječima. Tokom treninga, ELMo koristi kombinaciju unaprijed usmjerenog LSTM-a i unazad usmjerenog LSTM-a, gdje pokušava predvidjeti sljedeću riječ u rečenici na osnovu prethodnih riječi i prethodnu riječ na osnovu sljedećih riječi. Ovaj pristup se naziva dvosmjerno jezičko modelovanje (eng. *bidirectional language modeling*). Formalno funkcija cilja za unaprijed usmjereni model je:

$$\mathcal{L}_{unaprijed} = -\sum_{\mathrm{k=1}}^{\mathrm{N}} \log P(w_k \mid w_1, w_2, \ \ldots, w_{k-1}; \Theta), \tag{5.12}$$

a za unazad usmjereni model:

$$\mathcal{L}_{unazad} = -\sum_{k=1}^{N} \log P\,(w_k | w_{k+1}, w_{k+2}, \ldots, w_N; \Theta). \tag{5.13}$$

Tada je funkcija cilja koju ELMo minimizuje tokom treninga zbir prethodnih dviju funkcija:

$$\mathcal{L}_{\mathrm{ELMo}} = \mathcal{L}_{unaprijed} + \mathcal{L}_{\mathrm{unazad}}. \tag{5.14}$$

[9] https://www.washington.edu/

ELMo koristi višeslojnu arhitekturu, gdje se vektorske reprezentacije riječi generišu iz različitih slojeva mreže. Svaki sloj modela uči različite nivoe jezičkih informacija - niži slojevi hvataju osnovne morfološke i sintaksičke informacije, dok viši slojevi obuhvataju složenije semantičke odnose. Konačna vektorska reprezentacija svake riječi u ELMo-u se dobija kao linearna kombinacija vektora iz svih slojeva modela. Ova kombinacija je ponderisana na način da se težine mogu prilagoditi specifičnom NLP zadatku, što omogućava fleksibilnost i prilagodljivost modela.

Formalno, neka je w_i riječ u rečenici, a neka su $h_i^{(k)}$ vektori iz k-tog sloja BiLSTM mreže za tu riječ. Konačna ELMo reprezentacija riječi w_i, označena kao $\mathrm{ELMo}(w_i)$, računa se kao:

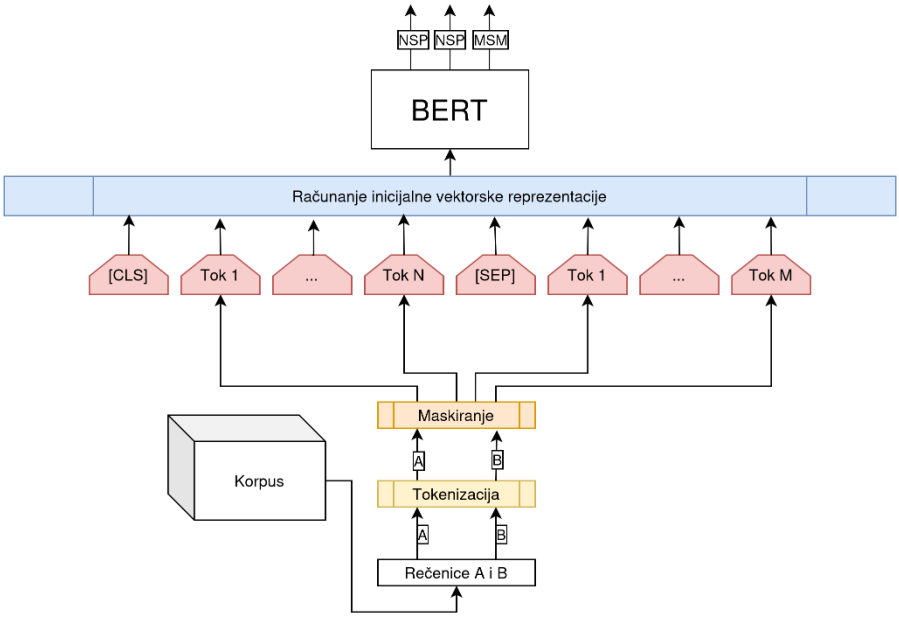
$$\mathrm{ELMo}(w_i) = \gamma \sum_{k=0}^{L} s_k h_i^{(k)}. \tag{5.15}$$

gdje je L ukupan broj slojeva, s_k ponder za k-ti sloj, a γ skalirajući faktor koji omogućava prilagođavanje veličine vektora. Nakon što je model treniran, ELMo ugrađivanje riječi za poziciju k u rečenici S dobija se kao kontekstualizovana reprezentacija koja kombinuje informacije iz svih slojeva LSTM mreže.

ELMo je pokazao poboljšanja u performansama na širokom spektru NLP zadataka u odnosu na prijašnje modele. Međutim, iako je ELMo bio revolucionaran u trenutku svog predstavljanja, ubrzo su se pojavili još napredniji modeli zasnovani na transformatorskim arhitekturama, kao što je BERT, koji je imao bolju sposobnost da razumije kontekstualne informacije.

5.3.2. BERT

BERT (*Bidirectional Encoder Representations from Transformers*) je kontekstualni jezički model razvijen od strane istraživača kompanije Google. BERT je baziran na arhitekturi transformatora, odnosno pošto se primarno primjenjuje za rješavanje problema klasifikacije, on sadrži isključivo enkoderski dio arhitekture, uz određene modifikacije u odnosu na prvobitnu arhitekturu. Pošto je prethodno objašnjena originalna arhitektura transformatora i mehanizam pažnje, u nastavku će fokus biti na specifične karakteristike i mehanizme koji čine BERT posebnim. Kao i ELMo, BERT je dvosmjerni model koji obrađuje tekst u oba smjera, dok se u prvobitnoj arhitekturi transformatora vršila jednosmjerna obrada uz maskiranje svih budućih tokena koji dolaze nakon posljednjeg generisanog tokena. Dvosmjernost se postiže tako što se BERT obučava da riješi dva zadatka, a to su maskirano jezičko modelovanje (eng. *masked language modeling* - MLM) i predviđanje naredne rečenice (eng. *next sentence prediction* - NSP). U pitanju je nenadgledano obučavanje [33-34]. Na slici 5.1 je prikazana arhitekturalna šema BERT-a.



Slika 5.1: Šematski prikaz pojednostavljene arhitekture BERT-a

Iz korpusa se u svakom koraku biraju dvije rečenice, gdje se u 50% slučajeva za rečenicu A bira rečenica B koja joj neposredno slijedi u tekstu, a u drugih 50% slučajeva bilo koja druga nasumično odabrana rečenica. Nakon toga se obje rečenice tokenizuju pomoću WordPiece tokenizacije, a onda se 15% tokena iz obje rečenice maskira. Maskiranje za date tokene se vrši tako što se u 80% slučajeva odabrani token zamjene sa tokenom [MASK], u 10% slučajeva se zamjeni sa nekim nasumičnim tokenom iz vokabulara, a u preostalih 10% se token ostavlja nepromjenjenim. Na početak sekvence se dodaje specijalni token [CLS], a između tokena prve rečenice i druge se dodaje specijalni token [SEP]. Sljedeći korak je kreiranje vektorske reprezentacije za sve tokene, gdje se za razliku od prvobitnog transformatora reprezentacija računa kao zbir pozicione reprezentacije koja se kod BERT-a mijenja tokom obučavanja, ali je kasnije fiksna kada se koristi tokom inferencije, inicijalne reprezentacije tokena i reprezentacije koja označava da li neki token pripada rečenici A ili B. BERT za svaki token t u ulaznoj sekvenci računa reprezentaciju $h_t \in \mathbb{R}^H$ nakon prolaska kroz višeslojni enkoder. Za MLM se na tu reprezentaciju primjenjuje linearni sloj, uz pripadajući vektor pomjeraja, formalno:

$$\boldsymbol{z}_t = \boldsymbol{h}_t W^{(MLM)} + \boldsymbol{b}^{(MLM)}, \quad \boldsymbol{W}^{(MLM)} \in \mathbb{R}^{H \times |V|}, \quad \boldsymbol{b}^{(MLM)} \in \mathbb{R}^{|V|}. \tag{5.16}$$

Nakon toga, nad z_i se primjenjuje softmaks funkcija kako bi se dobila vjerovatnoća za svaki token iz rječnika. Funkcija cijene se računa samo na maskiranim tokenima iz skupa M. Neka je ciljni token za poziciju i dat sa $y_i \in \{1, \dots, |V|\}$. Tada je funkcija cijene za MLM:

$$\mathcal{L}_{MLM} = -\sum_{i \in M} \log P\left(y_i \mid \boldsymbol{h_i}\right). \tag{5.17}$$

Za NSP zadatak se koristi latentna reprezentacija specijalnog tokena [CLS], označena kao $\boldsymbol{h}^{(CLS)} \in \mathbb{R}^H$. Na ovu reprezentaciju se primjenjuje linearni sloj sa pripadajućim pomjerajem:

$$\boldsymbol{z}_{NSP} = \boldsymbol{h}_{CLS} \boldsymbol{W}^{(NSP)} + \boldsymbol{b}^{(NSP)}, \quad \boldsymbol{W}^{(NSP)} \in \mathbb{R}^{H \times 2}, \quad \boldsymbol{b}^{(NSP)} \in \mathbb{R}^2. \tag{5.18}$$

Nakon toga se primjenjuje softmaks funkcija kako bi se dobila distribucija vjerovatnoća preko dvije klase. Ako je $s \in \{0,1\}$ indikator da li rečenica B zaista slijedi rečenicu A, tada se funkcija cijene za NSP računa koristeći unakrsnu entropiju za dvije klase:

$$\mathcal{L}_{NSP} = -[\, s \cdot \log P(NSP = 1) + (1 - s) \cdot \log P(NSP = 0)], \tag{5.19}$$

gdje su $P(\text{NSP} = 1)$ i $P(\text{NSP} = 0)$ vrijednosti dobijene iz softmaks distribucije nad $\boldsymbol{z}_{NSP}$.

Pošto se model istovremeno trenira za oba zadatka, ukupna funkcija cijene je suma:

$$\mathcal{L} = \mathcal{L}_{(MLM)} + \mathcal{L}_{(NSP)}. \tag{5.20}$$

Cilj obučavanja je minimizacija funkcije cijene $\mathcal{L}$.

BERT je pokazao bolje performanse u odnosu na ELMo i prethodno opisane modele, ali su se s vremenom pojavile još naprednije reprezentacije, poput onih razvijenih od kompanije OpenAI, od kojih je *text-embedding-3-small* korišten u praktičnom dijelu rada.

5.4. Evaluacija kvaliteta opisa

Evaluacija kvaliteta generisanih opisa je ključni korak u procjeni performansi modela. Cilj sistema za opisivanje slika je da generiše deskriptivne, informativne i koherentne tekstualne opise na osnovu vizuelnog sadržaja slike. Međutim, procjena kvaliteta ovih opisa predstavlja izazov, jer se zahtjeva balans između semantičke preciznosti, jezičke prirodnosti i usklađenosti sa referentnim opisima. U tu svrhu, koriste se različite metrike koje kvantitativno ocjenjuju performanse modela poređenjem generisanih opisa sa skupom referentnih opisa koje su kreirali ljudi. U nastavku su opisane najčešće korištene metrike pri evaluaciji.

5.4.1. BLEU

BLEU (*Bilingual Evaluation Understudy*) metrika je prvobitno razvijena za evaluaciju kvaliteta generisanog teksta u mašinskom prevođenju, ali je njena primjena brzo proširena i na druge oblasti, uključujući evaluaciju generisanih opisa slika. BLEU funkcioniše tako što upoređuje generisani opis slike sa jednim ili više referentnih opisa koje su napisali ljudi. Ključni koncept BLEU metrike je modifikovana preciznost n-grama, gdje n-gram predstavlja sekvencu n uzastopnih riječi [35-36].

Uobičajeno preciznost n-grama se računa kao broj n-grama iz generisanog opisa koji se pojavljuju u referentnim opisima podijeljen sa ukupnim brojem n-grama u generisanom opisu. Međutim, model može generisati opis, koji sadrži veliki broj ponavljanja „odgovarajućih“ riječi, te kao rezultat se dobija loš opis, koji ima visoku preciznost. Zbog toga BLEU uvodi modifikovanu preciznost n-grama kojom se izbjegava problem sa prekomjernim generisanjem riječi. To se postiže ograničavanjem broja pojavljivanja n-grama u generisanom opisu, sa brojem pojavljivanja tog n-grama u referentom opisu [35].

Primjer 5.3: Računanje unigram preciznosti i modifikovane unigram preciznosti.

Generisani opis 1: „crveni crveni crveni crveni.“

Generisani opis 2: „crveni automobil parkiran ispred garaže.“

Referentni opis: „crveni automobil parkiran ispred kuće.“

Uobičajena unigram preciznost za prvi generisani opis je $4/4$, a za drugi generisani opis je $4/5$, pa proizilazi da je prvi opis bolji nego drugi, što nije slučaj. Modifikovana unigram preciznost za prvi opis je $1/4$, a za drugi opis je $4/5$, odnosno ova metrika jasno pokazuje da je drugi opis bolji od prvog.

Modifikovanim pristupom se postiže prikladnost opisa, kada se koriste unigrami, a kada je $N \geq 2$, tada se postiže skladnost opisa. Prikladnost (eng. *adequacy*) u kontekstu generisanja opisa slika odnosi se na tačnost i kompletnost opisa slike. Opis je prikladan ako prenosi isti smisao i informacije kao referentni opis slike. Prikladnost se procjenjuje na osnovu toga koliko dobro generisani opis sadrži sve ključne informacije iz referentnog opisa. U kontekstu n-grama, upotreba istih riječi kao u referentnim opisima (unigrami) pomaže u postizanju prikladnosti jer osigurava da su osnovni pojmovi i informacije prisutni. Na primjer, ako slika prikazuje „crveni automobil parkiran ispred kuće," opis koji sadrži te ključne riječi bi se smatrao prikladnim. Skladnost (eng. *fluency*) se u kontekstu generisanja opisa slika odnosi na prirodnost i čitljivost generisanog opisa. Opis je skladan ako zvuči kao da ga je napisao izvorni govornik ciljanog jezika, bez gramatičkih grešaka i neprirodnih konstrukcija. Duži n-grami (poput bigrama, trigrama, itd.) se koriste za procjenu skladnosti jer duže sekvence riječi pomažu da se ocijeni koliko je opis prirodan i koliko tečno teče. Ako su duži n-grami u generisanom opisu slični onima u referentnim opisima, to znači da je generisani opis skladan i prirodan. Na primjer, umjesto „automobil crveni ispred kuće parkiran", skladan opis bi bio „crveni automobil parkiran ispred kuće." [35, 37].

Dok se predugi opisi već kažnjavaju jer uključuju dodatne n-grame koji se ne pojavljuju u referentnim opisima, kratki opisi mogu dobiti visoke ocjene, iako ne prenose dovoljno informacija kao duži opisi.

Da bi se izbjegla favorizacija prekratkih opisa, uvodi se kazna za kratkoću (eng. BP -*brevity penalty*). BP osigurava da generisani opisi budu dovoljno dugi da prenesu cjelokupni smisao referentnog opisa, što poboljšava sveukupnu ocjenu kvaliteta generisanog sadržaja. BP se primjenjuje kada je dužina generisanog opisa manja od dužine referentnog opisa. Matematički, BP se računa kao [35]:

$$BP = \begin{cases} 1 & \text{ako je } c > r \\ e^{(1-\frac{r}{c})}, & \text{ako je } c \leq r \end{cases}, \tag{5.21}$$

gdje je c dužina generisanog opisa, a r dužina referentnog opisa. U narednom primjeru je ilustrovana potreba za uvođenje BP. Konačni BLEU rezultat se računa kao:

$$BLEU = BP \cdot exp\left(\frac{1}{N}\sum_{n=1}^{N} log\, p_n\right), \tag{5.22}$$

gdje je N maksimalna dužina n-grama, a p_n modifikovana preciznost za n-grame dužine n.

Primjer 5.4: Računanje BLEU, za opise različite dužine.

Generisani opis 1: „Pas trči", dužina $c = 2$.

Generisani opis 2: „Pas trči i igra se u parku", dužina $c = 7$.

Referentni opis: „Crni pas trči i igra se u parku“, dužina $c = 8$.

Modifikovana unigram preciznost za prvi kandidatski opis je $p_1 = \frac{2}{2} = 1$, a za drugi kandidatski opis je $p_1 = \frac{7}{7} = 1$, pa prema tome ispada kako su oba opisa jednako dobra. Dalje se računa BP za oba opisa, gdje se za prvi kandidatski opis dobija da je $BP = e^{(1-8/2)} = e^{-3} \approx 0{,}05$, a za drugi kandidatski opis da je $BP = e^{(1-8/7)} = e^{-1/7} \approx 0{,}87$, konačno:

$$BLEU_1 = BP \cdot exp(log\, p_1) = 0{,}05 \cdot exp(log\, 1.0) = 0{,}05 \cdot 1 = 0{,}05, \qquad (5.23)$$

što je BLEU ocjena za prvo opis, a za drugi opis:

$$BLEU_2 = BP \cdot exp(log\, p_1) = 0{,}87 \cdot exp(log\, 1.0) = 0{,}87 \cdot 1 = 0{,}87, \qquad (5.24)$$

Sa niskim BP, prvi generisani opis će biti ozbiljno penalizovan zbog kratkoće, dok drugi generisani opis, iako ni on nije savršen, neće biti toliko penalizovan jer je duži i prenosi više informacija.

Glavna prednost BLEU metrike u odnosu na druge metrike je njena jednostavnost, zbog čega se brzo izračunava, a glavna mana je što ignoriše semantičku sličnost i sinonime [38].

5.4.2. ROUGE

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) je skup metrika koje se koriste za evaluaciju kvaliteta automatski generisanih tekstova, posebno u kontekstu sažimanja teksta i generisanja opisa slika. Iako postoji nekoliko varijanti ROUGE metrika, za evaluaciju tekstualnih opisa slika najčešće se koristi ROUGE-L varijanta [39-40].

ROUGE-L se zasniva na konceptu najdužeg zajedničkog podniza (eng. *longest common subsequence* - LCS) između generisanog teksta i referentnog teksta. Formalno, niz $Z = [z_1, z_2, \dots, z_n]$ je podniz drugog niza $X = [x_1, x_2, \dots, x_m]$ ako postoji strogo rastući niz $[i_1, i_2, \dots, i_k]$ indeksa niza X tako da za sve $j = 1, 2, \dots, k$ vrijedi $x_{i_j} = z_j$. Za dva niza X i Y, LCS niza X i Y je zajednički podniz sa maksimalnom dužinom [39].

Za procjenu sličnosti između referentnog opisa X dužine m i kandidatskog opisa Y dužine n, koristi se F-mjera bazirana na LCS, koja ujedno predstavlja i formulu za ROUGE-L [39]:

$$F_{\text{ROUGE-L}} = \frac{(1+\beta^2) \cdot R_{\text{LCS}} \cdot P_{\text{LCS}}}{R_{\text{LCS}} + \beta^2 \cdot P_{\text{LCS}}}, \qquad (5.25)$$

gdje su R_{LCS} i P_{LCS} definisani kao:

$$R_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{m}, \qquad (5.26)$$

$$P_{\text{LCS}} = \frac{\text{LCS}(X,Y)}{n} \qquad (5.27)$$

i predstavljaju odziv i preciznost bazirani na LCS, a $LCS(X,Y)$ je dužina LCS između X i Y. Konačno β je hiperparametar kojim se balansira između važnosti odziva i preciznosti. Obično se uzima da je $\beta = 1$, kako bi dala jednaka važnost odzivu i preciznosti [39].

Primjer 5.5: Računanje ROUGE-L za referentne opise različitih dužina.

Referentni opis, R: „Crni pas trči i igra se u parku“; dužina $m = 8$.

Generisani opis 1, G_1: "Pas trči", dužina $n_1 = 2$.

Generisani opis 2, G_2: "Pas trči i igra se u parku", dužina $n_2 = 7$.

Generisani opis 3, G_3: "Crni pas trči brzo i igra se u parku sa crvenom teniskom loptom", dužina $n_3 = 12$.

Za prvi generisani opis je $\mathrm{LCS}(R, G_1) = 2$, za drugi je $\mathrm{LCS}(R, G_2) = 7$, a za treći je $\mathrm{LCS}(R, G_3) = 7$. Dalje su odziv i preciznost za prvi generisani opis $R_{\mathrm{LCS}} = \frac{2}{8} = 0{,}25$, $P_{\mathrm{LCS}} = \frac{2}{2} = 1{,}0$. Za drugi $R_{\mathrm{LCS}} = \frac{7}{8} = 0{,}875$, $P_{\mathrm{LCS}} = \frac{7}{7} = 1{,}0$. Za treći $R_{\mathrm{LCS}} = \frac{7}{8} = 0{,}875$, $P_{\mathrm{LCS}} = \frac{7}{12} = 0.5833$. Neka je $\beta = 1$, tada je:

$$F_{\text{ROUGE-L}} = \frac{(1+1) \cdot 0{,}25 \cdot 1{,}0}{0{,}25 + 1} = \frac{2 \cdot 0{,}25}{1{,}25} = \frac{0{,}5}{1{,}25} = 0{,}4. \tag{5.28}$$

ROUGE-L ocjena za prvi generisani opis, za drugi:

$$F_{\text{ROUGE-L}} = \frac{(1+1) \cdot 0{,}875 \cdot 1{,}0}{0{,}875 + 1} = \frac{2 \cdot 0{,}875}{1{,}875} = \frac{1{,}75}{1{,}875} \approx 0{,}933, \tag{5.29}$$

te za treći:

$$F_{\text{ROUGE-L}} = \frac{(1+1) \cdot 0{,}875 \cdot 0{,}5833}{0{,}875 + 0{,}5833} = \frac{1{,}75 \cdot 0{,}5833}{1{,}4583} = \frac{1{,}0208}{1{,}4583} \approx 0{,}70. \tag{5.30}$$

Prvi generisani opis, ima nižu ocjenu, jer je kratak i nedovoljno sličan sa referentnim opisom. S druge strane, drugi generisani opis ima visoku ocjenu, jer je slične dužine kao referentni opis i ima veću strukturalnu sličnost sa referentnim opisom. Treći generisani opis, iako je duži i detaljniji, ima nižu ocjenu, zbog dodatnih riječi koje nisu dio LCS, što smanjuje preciznost.

Prednost ROUGE-L u odnosu na druge ROUGE varijante, ali i u odnosu na BLEU, je to što uzima u obzir redoslijed riječi pomoću LCS, što omogućava bolje hvatanje strukturalne sličnosti između generisanog i referentnog opisa. Takođe, ROUGE-L ne zahtjeva predefinisanje dužine n-gramova, što ga čini fleksibilnijim za različite tipove opisa. Međutim, kao i BLEU, ROUGE-L ne uzima u obzir semantičku sličnost riječi. Dodatno, ROUGE-L može biti osjetljiv na dužinu opisa, što može dovesti do nepravednih ocjena za kraće ili duže opise u poređenju sa referentnim opisom, kao što je pokazano na primjeru 5.5. To je posebno nepovoljno u situacijama kada je kraći opis adekvatan ili kada bi duži opis pružio dodatne informacije od značaja koje su izostavljene u referentnom opisu.

5.4.3. METEOR

METEOR (*Metric for Evaluation of Translation with Explicit ORdering*) je metrički sistem koji je prvobitno razvijen za ocjenjivanje mašinskog prevođenja, ali je kasnije proširen i na druge oblasti kao što je opisivanje slika pomoću teksta. METEOR je uveden kao alternativa BLEU metrici zbog njene rigidnosti i nedostatka fleksibilnosti u prepoznavanju sinonima i sličnih struktura rečenica [38].

METEOR kreira poravnanje između dva opisa, gdje svaka riječ (unigram) iz jednog opisa može biti mapirana na nula ili jednu riječ iz drugog opisa, ali nikada više od jedne riječi. Proces poravnanja se vrši u etapama, a svaka etapa ima dvije faze. U prvoj fazi, eksterni modul izlistava sve moguće parove unigrama između dva opisa. Na primjer, ako se riječ "računar" pojavljuje jednom u kandidatskom opisu, a dva puta u referentnom, modul će kreirati dva moguća mapiranja, jedno mapiranje prema prvom pojavljivanju u referentnom opisu, a jedno ka drugom. Različiti moduli koriste različite kriterijume za mapiranje, te od broja modula zavisi i broj etapa. Uobičajeno se koriste tri modula, prvi modul mapira identična podudaranja, drugi modul mapira riječi koje su iste nakon što se korjenuju[10] (eng. *stemming*), a treći modul mapira riječi koje su sinonimi[11]. U drugoj fazi svake etape bira se najveći podskup ovih mapiranja koji čini poravnanje. Ako postoji više takvih podskupova sa istim brojem mapiranja, METEOR bira onaj sa najmanje ukrštanja mapiranja. Formalno, dva mapiranja unigrama (t_i, r_j) i (t_k, r_l), gdje su t_i i t_k unigrami u generisanom opisu mapirani na unigrame r_j i r_l u referentnom opisu, redom, se smatraju ukrštenim ako i samo ako sljedeća formula daje negativan rezultat:

$$(pos(t_i) - pos(t_k)) \times (pos(r_j) - pos(r_l)), \tag{5.31}$$

gdje je $pos(t_x)$ numerička pozicija unigrama t_x u nizu sistemskog prevoda, a $pos(r_y)$ je numerička pozicija unigrama r_y u referentnom nizu.

Nakon što se izvrše sve faze, računa se METEOR rezultat. Preciznost P se računa kao odnos broja mapiranih unigrama u generisanom opisu prema ukupnom broju unigrama u generisanom opisu, dok se odziv R računa kao odnos broja mapiranih unigrama prema ukupnom broju unigrama u referentnom opisu. Zatim se računa harmonijska sredina, odnosno F-mjera:

$$F_\alpha = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R}, \tag{5.32}$$

gdje je α hiperparametar, kojim se balansira između važnosti preciznosti i odziva. Uobičajeno se uzima vrijednost $\alpha = 0{,}9$, kako bi se dao veći značaj odzivu [38].

METEOR uvodi kazne za neusaglašenost reda riječi u generisanom opisu kako bi se kaznili opisi koji sadrže neprirodne konstrukcije. Kazna je definisana kao:

$$Kazna = \gamma \cdot \left(\frac{C}{M}\right)^{\delta}, \tag{5.33}$$

gdje je C broj grupa riječi koje nisu u odgovarajućem redoslijedu, M ukupan broj poravnatih riječi, a γ i δ su hiperparametri koji kontrolišu jačinu penalizacije. Tipične vrijednosti su $\gamma = 0{,}5$ i $\delta = 3{,}0$.

Konačno METEOR se računa kao:

[10] Korjenovanje je proces kojim se riječi svode na njihov korijen ili osnovni oblik uklanjanjem nastavaka i prefiksa, čime se smanjuje broj jedinstvenih oblika riječi.

[11] *WordNet* je leksička baza podataka u kojoj su riječi organizovane u skupove sinonima (eng. *synsets*) koji predstavljaju pojmove sličnog značenja, te se koriste za pronalazak riječi koje, iako nisu identične, imaju istu ili sličnu semantičku vrijednost.

$$\text{METEOR} = F \cdot (1 - \text{Kazna}). \tag{5.34}$$

Primjer 5.6: Računanje METEOR ocjene za različite generisane opise, pod pretpostavkom da su uzete uobičajene vrijednosti za prethodno pomenute parametre.

Referentni opis: "Mali crni pas trči za žutim frizbijem u gradskom parku.", broj unigrama $r = 10$.

Generisani opis 1: "Mali smeđi pas trči za frizbijem u parku.", broj unigrama $n = 8$.

Generisani opis 2: "Mali crni ker trči za žutim frizbijem u parku.", broj unigrama $n = 9$.

Generisani opis 3: "Crni trči žutim za pas gradskom frizbijem mali parku u.", broj unigrama $n = 10$.

Na slici 5.2 su prikazani rezultati poravnanja generisanih opisa sa referentnim opisom.

C = 4	1	2	3	4	5	6	7	8	9	10
Ref	mali	crni	pas	trči	za	žutim	frizbijem	u	gradskom	parku
Gen 1	mali	smeđi	pas	trči	za	frizbijem	u	parku		

C = 2	1	2	3	4	5	6	7	8	9	10
Ref	mali	crni	pas	trči	za	žutim	frizbijem	u	gradskom	parku
Gen 2	mali	crni	ker	trči	za	žutim	frizbijem	u	parku	

C = 10	1	2	3	4	5	6	7	8	9	10
Ref	mali	crni	pas	trči	za	žutim	frizbijem	u	gradskom	parku
Gen 3	crni	trči	žutim	za	pas	gradskom	frizbijem	mali	parku	u

Slika 5.2: METEOR poravnanje

Za prvi generisani opis ukupno je poravnato sedam riječi $m = 7$, a nakon poravnanje izdvajaju se četiri segmenta $C = 4$. Pa je preciznost $P = \frac{m}{n} = \frac{7}{8} = 0{,}875$, a odziv $R = \frac{m}{r} = \frac{7}{10} = 0{,}7$, pa je $F \approx 0{,}788$. Konačno $METEOR = 0{,}715$.

Za drugi generisani opis ukupno je poravnato devet riječi $m = 9$, pošto su riječi "pas" i "ker" sinonimi u srpskom jeziku, a nakon poravnanja se izdvajaju dva segmenta $C = 2$. Pa je preciznost $P = 1$, a odziv $R = 0{,}9$, a $F = 0{,}953$. Konačno $METEOR \approx 0{,}949$.

Treći generisani opis sadrži sve iste riječi kao i referentni opis $m = 10$, ali je skroz pogrešan redoslijed riječi u rečenici pa je za poravnanje potrebno formirati deset segmenata $C = 2$. Preciznost, odziv i F-mjera su jedan $P = R = F = 1$, te da ne postoji kazna za neusklađenost u odnosu na referentni opis, ispalo bi da je treći generisani opis savršen, ovako je zbog kazne $METEOR = 0{,}5$.

5.4.4. CIDEr

CIDEr (*Consensus-based Image Description Evaluation*) je metrika razvijena kako bi poboljšala procjenu kvaliteta automatski generisanih opisa slika i bila u skladu sa ljudskim ocjenama, oslanjajući se na semantički značaj i važnost pojedinih riječi i fraza. Za razliku od metrika BLEU i ROUGE, koje uglavnom prate tačno poklapanje n-grama, te METEOR metrike, koja uvodi osnovnu podršku za sinonime i djelimična poklapanja, CIDEr ide korak dalje koristeći TF-IDF ponderisanje i kosinusnu sličnost kako bi prepoznao rijetke, ali bitne riječi, kao i različite sinonime koje prenose istu ideju, čime se postiže veća osjetljivost na riječi

koje su važne za datu sliku, a manje pažnje se posvjećuje učestalim ili generičkim riječima [41].

CIDEr računa sličnost između kandidatskog opisa C_i i referentnih opisa S_j (gdje je $j = 1, 2, \dots, m$), i to za n-grame dužine od 1 do N. U praksi se često uzimaju n-grami dužine od 1 do 4, čime se osigurava balans između osnovnih i složenijih jezičkih obrazaca. Konačni rezultat je prosječna vrijednost ovih sličnosti za sve dužine n-grama. Za sve generisane i referentne opise izdvajaju se n-grami određene dužine n, gdje se svaki opis predstavlja kao vektor TF-IDF težina n-grama. Formalno, ako je G skup svih mogućih n-grama u korpusu, tada je svaki opis o predstavljen kao vektor:

$$v_o = \left[w_o(g_1), w_o(g_2), \dots, w_o\left(g_{|G|}\right)\right], \tag{5.35}$$

gdje je $w_o(g_i) = \text{TF-IDF}(g_i, o)$ težina za n-gram g_i u opisu o. Nakon toga, računa se kosinusna sličnost između vektora generisanog opisa c i svakog vektora referentnog opisa r. CIDEr vrijednost za datu dužinu n-grama za generisani opis c računa se kao prosjek sličnosti sa svim referentnim opisima:

$$\text{CIDEr}_n(c, \{r_1, r_2, \dots, r_m\}) = \frac{1}{m}\sum_{i=1}^{m} \text{sim}_n(c, r_i), \tag{5.36}$$

gdje $\text{sim}_n(c, r_i)$ predstavlja kosinusnu sličnost između vektora c i r_i za n-grame dužine n. Konačna CIDEr vrijednost za generisani opis c računa se kao prosjek vrijednosti $\text{CIDEr}_n(c)$ za sve dužine n-grama od 1 do N:

$$\text{CIDEr}(c, \{r_1, r_2, \dots, r_m\}) = \frac{1}{N}\sum_{n=1}^{N} \text{CIDEr}_n(c, \{r_1, r_2, \dots, r_m\}). \tag{5.37}$$

Zahvaljujući TF-IDF ponderima, CIDEr prepoznaje važnost specifičnih termina koji su ključni za opis slike. Empirijski je potvrđeno da CIDEr postiže veću usklađenost s ljudskim ocjenama u poređenju sa starijim metrikama poput BLEU i ROUGE, a često i u odnosu na METEOR [41]. Međutim, da bi CIDEr davao relevantne i pouzdane ocjene, ključno je da postoji dovoljan broj različitih referentnih opisa za svaku sliku, jer veća raznolikost referenci omogućava preciznije ocjenjivanje semantičke sličnosti.

5.4.5. SPICE

SPICE (*Semantic Propositional Image Caption Evaluation*) je metrika razvijena s ciljem da bolje mjeri semantičku sličnost između referentnih i kandidatskih opisa. Dok su BLEU, ROUGE, METEOR i CIDEr zasnovani na n-gram poklapanjima, uz različite stepene ugrađene tolerantnosti na leksičke ili sintaksičke varijacije, SPICE uvodi modelovanje značenja na nivou objekata i njihovih međusobnih odnosa. Na taj način postiže se mjerenje semantičke usklađenosti opisa, bez obzira na to kako su rečenice sročene, što uglavnom nije slučaj sa prethodno opisanim metrikama [42].

Za računanje SPICE ocjene se formira scenski graf (eng. *scene graph*) gdje se identifikuju objekti, atributi i relacije između objekata. Scenski graf za neki opis d definisan je trojkom:

$$G(d) = \langle O(d), E(d), K(d) \rangle, \tag{5.38}$$

gdje je $\mathrm{O}(d) \subseteq C$ skup objekata pomenutih u opisu, $\mathrm{E}(d) \subseteq \mathrm{O}(d) \times R \times O(d)$ skup grana koje predstavljaju relacije između objekata, a $\mathrm{K}(d) \subseteq O(d) \times A$ skup atributa povezanih sa objektima. Formalno, neka je dat skup referentnih opisa $S = \{s_1, s_2, \dots, s_m\}$ i kandidatski opis c. Scenski graf za kandidatski opis označava se kao $G(c)$, dok se scenski graf za skup referentnih opisa S označava sa $G(S)$, a on se formira unijom scenski grafova svih referentnih opisa $G(s_i)$ gdje $s_i \in S$, uz kombinovanje sinonimnih čvorova.

Za potrebe evaluacije, scenski graf se transformiše u skup n-torki koje izražavaju semantičke iskaze izvedene iz objekata, atributa i relacija. Neka je data funkcija T koja generiše n-torke iz scenskog grafa:

$$T\big(G(c)\big) \triangleq \{(o) \mid o \in O(c)\} \cup \{(o,a) \mid (o,a) \in K(c)\} \cup \{(o_1, r, o_2) \mid (o_1, r, o_2) \in E(c)\} \tag{5.39}$$

gdje n-torke mogu sadržati jedan, dva ili tri elementa, u zavisnosti od toga da li predstavljaju objekte, objekte i atribute ili relacije između dva objekta. SPICE metrika upoređuje skupove n-torki iz scenski grafova kandidatskog opisa $T\big(G(c)\big)$ i referentnih opisa $T\big(G(S)\big)$. Tada se preciznost, odziv i SPICE definišu kao:

$$P(c,S) = \frac{\left|T\big(G(c)\big) \otimes T\big(G(S)\big)\right|}{\left|T\big(G(c)\big)\right|}, \tag{5.40}$$

$$R(c,S) = \frac{\left|T\big(G(c)\big) \otimes T\big(G(S)\big)\right|}{\left|T\big(G(S)\big)\right|}, \tag{5.41}$$

$$SPICE(c,S) = F1(c,S) = \frac{2 \cdot P(c,S) \cdot R(c,S)}{P(c,S) + R(c,S)}. \tag{5.42}$$

Gdje $\otimes$ operator vraća podudaranja[12] između n-torki iz dva scenska grafa.

Primjer 5.7: Računanje SPICE metrike za sliku 5.3 za koju je dato pet referentnih opisa i jedan kandidatski opis.

Slika 5.3: Slika psa za SPICE primjer

Referentni opisi:

1. Pas trči po plitkoj vodi.

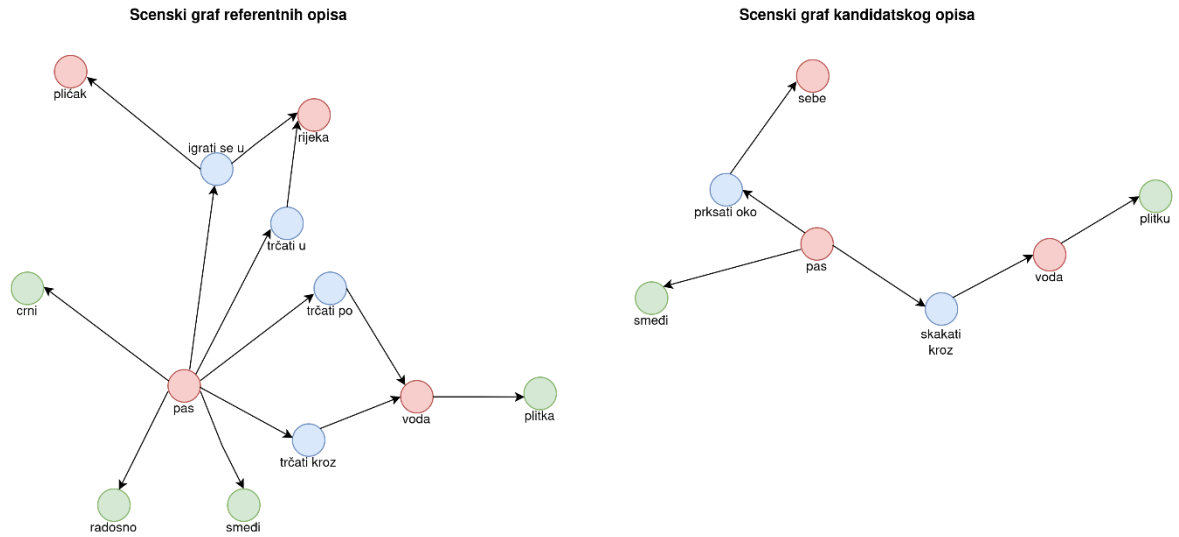
[12] Za podudaranje se gledaju i sinonimi iz *WordNet-a*.

2. Crni pas se igra u rijeci.
3. Crni pas trčkara u rijeci.
4. U plićaku se igra tamno smeđi ker.
5. Smeđi pas se radosno igra jureći kroz vodu.

Kandidatski opis:

1. Pas skače kroz vodu, prskajući oko sebe.

Na osnovu opisa formiraju se scenski grafovi, kao što je ilustrovano na slici 5.4.



Slika 5.4: Scenski grafovi referentnih i kandidatskog opisa

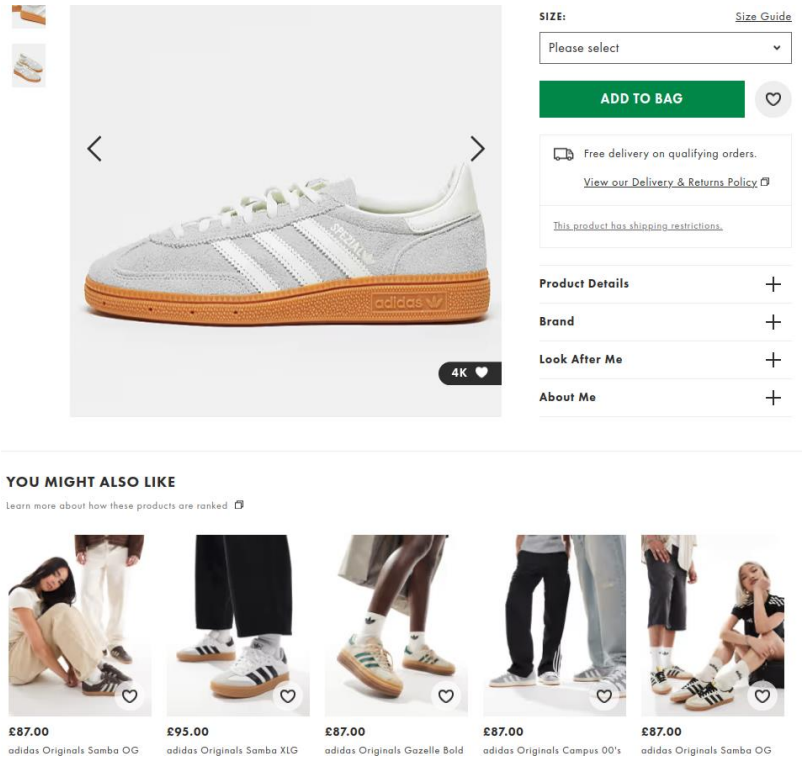
Sinonimi poput "pas" i "ker" u grafu su svedeni na jedan čvor kao što je ranije pomenuto. Za referentne opise postoji trinaest n-torki na osnovu grafa: *{(pas), (plićak), (rijeka), (voda), (pas, radosno), (pas, smeđi), (pas, crni), (voda, plitka), (pas, trčati kroz, voda), (pas, trčati po, voda), (pas, igrati se u, plićak), (pas, igrati se u, rijeka), (pas, trčati u, rijeka)}*, a za kandidatski opis ih je sedam: *{(pas), (sebe), (voda), (pas, smeđi), (voda, plitku), (pas, prskati oko, sebe), (pas, skakati kroz, voda)}*, pri čemu postoje četiri preklapanja, pa je preciznost $P = 4 / 7 \approx 0{,}57$, a odziv $R = 4 / 13 \approx 0{,}31$, konačno je $SPICE \approx 0{,}40$.

Glavna mana SPICE metrike je složenost njenog izračunavanja, ali i to što je SPICE zamišljen da mjeri semantiku, odnosno šta se kaže, a ne kako se kaže. Stoga može visoko ocijeniti iskaz koji je semantički „tačan“, ali gramatički loš ili potpuno nepravilan [42]. Zbog toga je uvedena SPIDER metrika [43], gdje se uzima prosječna vrijednost CIDEr-a i SPICE-a, da bi se pri evaluaciji opisa slika, ocijenila i semantička korektnost, ali i leksička, odnosno n-gram pokrivenost. Na taj se način izbjegava situacija da opis dobije visoku ocjenu samo zato što sadrži tačne ključne riječi (CIDEr), ili samo zato što je semantički ispravno prepoznao odnose među objektima (SPICE), a zanemario jezičku primjenu.

6. PRETRAGA PRODAVNICE

Pretraga prodavnice predstavlja postupak pronalaženja proizvoda unutar prodavnice, pri čemu je cilj pronalazak onih artikala koji bi bili od interesa za krajnjeg korisnika. U tom smislu, pretraga se može vršiti na dva osnovna načina: tekstualnom pretragom i vizuelnom pretragom. U prvom slučaju, korisnik kao upit unosi ključne riječi ili nazive proizvoda pomoću standardnog pretraživačkog alata, dok se u drugom slučaju kao upit koristi slika određenog proizvoda ili slika neke osobe koja nosi određenu odjevnu kombinaciju, a koja je od interesa krajnjem korisniku. Oba modaliteta imaju za cilj da pruže relevantne rezultate, ali se razlikuju u načinu ekstrakcije informacija i interpretaciji korisničkog unosa.

Ovaj rad se bavi problemima koji se javljaju kod pretrage proizvoda odjevnih predmeta, kada se koristi vizuelni modalitet, odnosno kada je upit slika. Potrebno je i naglasiti da ne mora korisnik eksplicitno da otpočne pretragu drugih proizvoda, već se pretraga implicitno vrši i kada korisnik već razgleda neki proizvod, jer se često ispod dijela koji prikazuje osnovne informacije o proizvodu prikazuje i lista sličnih i povezanih proizvoda za dati proizvod, a koja je dobijena primjenom tehnika za pretragu prodavnice, gdje je kao upit korištena slika trenutno razmatranog proizvoda, kao što je ilustrovano na slici 6.1. Za svrhe istraživanja u ovom radu nije bitno da li je riječ o eksplicitnoj ili implicitnoj pretrazi, zbog čega se u nastavku rada neće praviti distinkcija između ova dva pristupa.

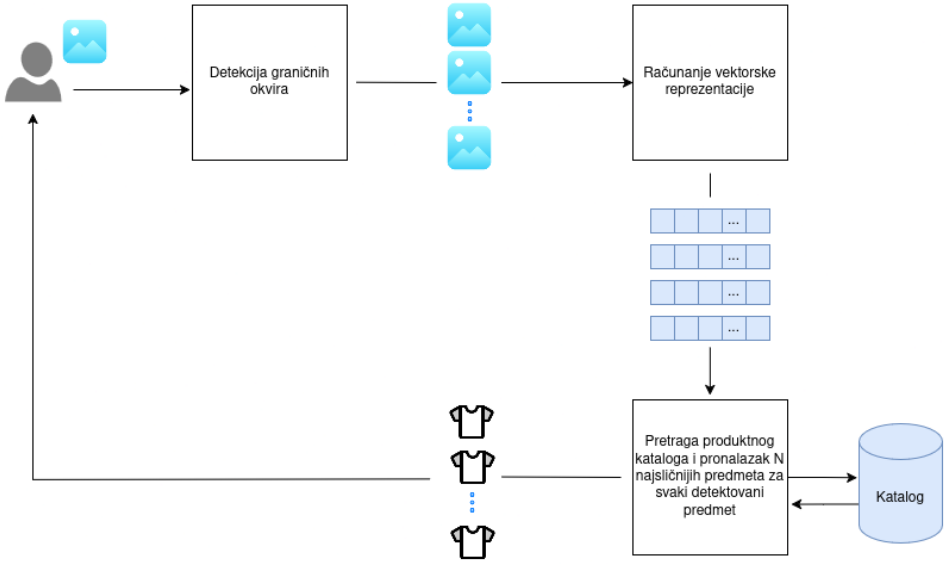


Slika 6.1 Primjer primjene pretrage prodavnice za pronalazak sličnih proizvoda

U daljnjem tekstu rada, prvo je dat formalan opis tradicionalnog postupka pretrage prodavnice, nakon toga su nabrojani najčešći nedostaci ovog pristupa. Potom je detaljno opisano predloženo rješenje, bazirano na primjeni opisivanja slika pomoću tehnika mašinskog učenja u tandemu sa postojećim pristupom.

6.1. Opis problema

Primjer tradicionalnog pristupa za pretragu prodavnice je ilustrovan na slici 6.2. Prvi korak uključuje detekciju graničnih okvira na ulaznoj slici I, gdje se koristi napredni algoritam detekcije objekata, poput YOLO[13] ili Mask R-CNN, da identifikuje skup graničnih okvira $\mathrm{B} = \{\mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_k\}$ pri čemu svaki okvir b_i definiše koordinate (x, y, w, h) [44] [45]. Nakon identifikacije graničnih okvira, za svaki okvir b_i računa se vektorska reprezentacija v_i primjenom konvolucione neuronske mreže (eng. *convolutional neural network*), rezultujući skupom vektora $V = \{v_1, v_2, \ldots, v_k\}$ gdje je $v_i \in \mathbb{R}^d$. Sljedeći korak uključuje pretragu kataloga proizvoda $C = \{c_1, c_2, \ldots, c_m\}$, gdje svaki proizvod c_j ima svoju vektorsku reprezentaciju. Za svaki vektor v_i izračunava se sličnost sa proizvodima c_j korišćenjem metričke funkcije, kao što su kosinusna sličnost ili Euklidska udaljenost, čime se identifikuje skup N najsličnijih proizvoda $S_i = \{s_{i1}, s_{i2}, \ldots, s_{iN}\}$. Konačno, korisniku se prikazuje skup preporuka $S = \{S_1, S_2, \ldots, S_k\}$ [4-5].



Slika 6.2: Šematski prikaz arhitekture za tradicionalni pristup pretrage prodavnice

6.1.1. Obučavanje modela za detekciju objekata

Neka je dat skup podataka $D = \{(x_i, y_i)\}_{i=1}^{N}$, gdje je $x_i \in \mathbb{Z}^{H \times W \times K}$ slika, a y_i pripadajuće oznake za tu sliku. Svaka oznaka y_i sastoji se od skupa graničnih okvira i odgovarajućih klasa za tu sliku. Ako se na slici x_i nalazi n_i objekata garderobe, tada je:

$$y_i = \{(b_{i1}, c_{i1}), (b_{i2}, c_{i2}), \ldots, (b_{in_i}, c_{in_i})\},$$

gdje je $b_{ij} = (x_{min}^{ij}, y_{min}^{ij}, x_{max}^{ij}, y_{max}^{ij})$ skup koordinata koje definišu granični okvir objekta j na slici i, a $c_{ij} \in C$ je klasa tog objekta (npr., "majica", "haljina", "pantalone" itd.). Cilj je obučiti model detekcije objekata f_θ, koji mapira ulaznu sliku x_i u skup predviđenih graničnih okvira i klasa:

$$\hat{y}_i = \{(\widehat{b_{ik}}, \widehat{c_{ik}})\}_{k=1}^{\widehat{n_i}},$$

[13] https://docs.ultralytics.com/

gdje je $\widehat{n_i}$ broj predviđenih objekata na slici x_i, $\widehat{b_{ik}}$ su predviđeni granični okviri, a $\widehat{c_{ik}}$ su predviđene klase. Model f_θ se trenira da minimizuje ukupnu funkciju cijene $L(\theta)$, koja kvantifikuje razliku između predviđanja modela i stvarnih oznaka. Funkcija cijene sastoji se iz dva dijela:

- **Cijena klasifikacije** (L_{cls}): Mjera greške u predviđanju klasa objekata. Obično se koristi unakrsna entropija (eng. *cross-entropy loss*) za više klasa.
- **Cijena regresije** (L_{reg}): Mjera greške u predviđanju koordinata graničnih okvira. Često se koristi *Smooth* L_1 ili L_2 norma za regresiju koordinata.

Ukupna funkcija cijene je data izrazom:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left(L_{\text{cls}}\left(c_{ij}, \widehat{c_{ij}}\right) + \lambda L_{\text{reg}}\left(b_{ij}, \widehat{b_{ij}}\right)\right),$$

gdje je N broj slika u skupu podataka, n_i broj stvarnih objekata na slici x_i , λ je hiperparametar koji balansira uticaj između funkcije cijene klasifikacije i funkcije cijene regresije, c_{ij} i $\widehat{c_{ij}}$ su stvarne i predviđene klase, a b_{ij} i $\widehat{b_{ij}}$ su stvarni i predviđeni granični okviri.

6.1.2. Obučavanje modela za analizu sličnosti slika

Nakon detekcije objekata i izdvajanja pojedinačnih predmeta garderobe sa slika, cilj je obučiti model koji može da mapira ove predmete u vektorski prostor gdje su više slični predmeti bliži jedni drugima, a manje slični predmeti udaljeniji jedni od drugih. Neka je dat model g_ϕ, koji mapira ulaznu sliku predmeta x_{ij} (određenu graničnim okvirom b_{ij}) u vektorsku reprezentaciju $z_{ij} \in \mathbb{R}^d$, odnosno $z_{ij} = g_\phi(x_{ij})$.

Obučavanje modela g_ϕ zasniva se na korišćenju funkcije cijene tipa *triplet loss* zajedno sa strategijom *hard mining* [5]. Formiraju se trojke (a, p, n), gdje su:

- a sidro (eng. *anchor*), slika nekog predmeta,
- p pozitivan uzorak, slika istog predmeta,
- n negativni uzorak, slika različitog predmeta.

Nakon što se svaka od tih slika proslijedi kroz model g_ϕ, dobijaju se vektorske reprezentacije $z_a = g_\phi(a), z_p = g_\phi(p)$ i $z_n = g_\phi(n)$. *Triplet loss* funkcija definiše se kao:

$$L_{\text{triplet}}(\phi) = \sum_{(a,p,n)}\left[|z_a - z_p|_2^2 - |z_a - z_n|_2^2 + \alpha\right]_+,$$

gdje je $\alpha > 0$ margina koja definiše minimalnu razliku između pozitivnih i negativnih parova, kako ne bi bili suviše blizu jedan drugom, a $[x]_+ = \max(0, x)$ se koristi u funkciji cijene kako bi se osiguralo da samo pozitivni tj. problematični slučajevi doprinesu cijeni, dok se negativni zanemaruju.

Hard mining strategija podrazumijeva izbor najtežih pozitivnih i negativnih uzoraka tokom obučavanja. Za svako sidro a, bira se pozitivni primjer p koji je najudaljeniji od a u trenutnom

vektorskom prostoru, što znači da ga model teže razlikuje od negativnih. Za svako sidro a, bira se negativni primjer n koji je najbliži a, tj. onaj koji je za model najteže razlikovati od pozitivnih. Kroz iterativno ažuriranje parametara ϕ, model uči vektorske reprezentacije koje bolje odražavaju semantičku sličnost između predmeta. Na ovaj način, model je sposoban da kodira predmete garderobe tako da su slični predmeti (npr., ista majica fotografisana u različitim kontekstima) blizu u vektorskom prostoru, dok su različiti predmeti udaljeni.

Za potrebe rada dotrenirana (eng. *fine-tuned*) su tri modela za detekciju objekata: model baziran na Detectron2 Mask-RCNN, YOLOv8 i YOLOv11 nad *DeepFashion2*[14] - *DF2* skupu podataka. Pored toga, obučen je jedan model za potrebe analize sličnosti slika baziran na ResNet50 CNN nad istim skupom podataka. DF2 je skup podataka namijenjen za istraživanje u oblasti računarskog vida i modne industrije. Sadrži slike odjevnih predmeta s detaljnim oznakama, uključujući granične okvire, maske segmentacije (eng. *segmentation masks*), atribute odjeće i informacije o kategorijama. DF2 uključuje 192.000 slika za obuku, 32.000 za validaciju i 63.000 za testiranje.

6.1.3. Nedostaci tradicionalnog pristupa za pretragu proizvoda

Razlikuju se dvije vrste nedostataka kod tradicionalnog pristupa za pretragu proizvoda, one koje potiču od modela za detekciju objekata i one koje potiču od modela za analizu sličnosti slika.

6.1.3.1. Nedostaci modela za detekciju objekata

Pošto je prvi korak u sistemu pretrage prodavnice detekcija odjevnih predmeta i njihovih graničnih okvira, greške u ovoj fazi mogu nepovratno poremetiti čitav proces. Najčešće greške su [46]:

- Pogrešno određen granični okvir. Dešava se kada model pogrešno odredi granice okvira za neki odjevni predmet, pa se može desiti da se ne pronađu dovoljno slični predmeti.
- Pogrešna klasifikacija graničnog okvira. Dešava se kada model dodijeli pogrešnu klasu nekom graničnom okviru. Moguće je da je okvir pravilno određen, ali da se zbog pogrešne klase ne pronađu dovoljno slični predmeti.

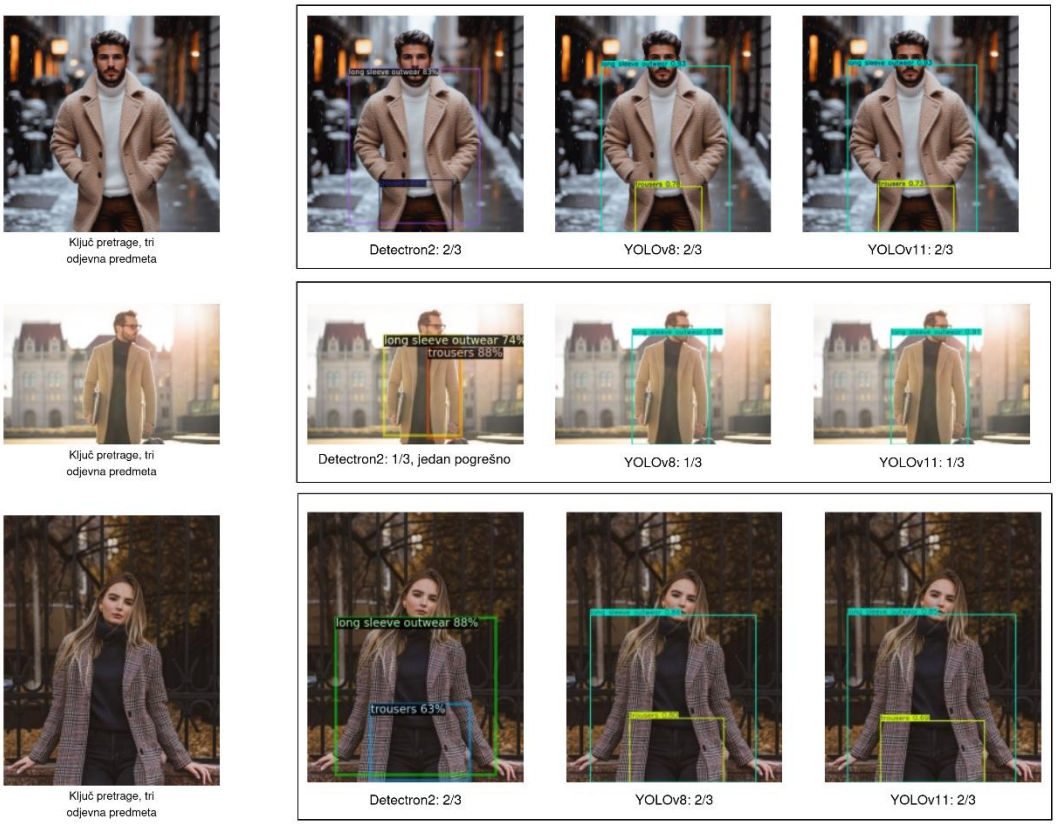
Prethodna dva problema se mogu uglavnom prevazići analizom pogrešnih uzoraka i proširivanjem trening skupa. Ipak, postoje i greške koje se ne mogu prevazići na ovakav način:

- Detekcija lažno pozitivnih uzoraka. Ako je upit majica na kojoj su naslikani ljudi, model će detektovati granične okvire za garderobu koju ti naslikani ljudi nose, što rezultuje velikim brojem lažno pozitivnih uzoraka. Primjer detekcije lažno pozitivnih uzoraka prikazan je na slici 6.3.

[14] https://github.com/switchablenorms/DeepFashion2

Slika 6.3: Primjer detekcije lažno pozitivnih uzoraka, kada je na majici prikazan neki broj ljudi

- Detekcija slojevite odjeće. Slojevita garderoba predstavlja složen upit i predstavlja veliki problem za modele za detekciju. Kako su određeni slojevi odjeće samo djelimično vidljivi, model ih "ne vidi" i obično se u takvim situacijama samo detektuje posljednji sloj garderobe koji je i najvećim dijelom vidljiv. Ovo je problem u situacijama kada potrošač želi u potpunosti da rekreira stil garderobe iz upita. Primjer problema sa slojevitom garderobom je prikazan na slici 6.4, gdje od vidljive garderobe ni na jednoj osobi nije detektovan džemper.



Slika 6.4: Primjer problema sa detekcijom sve odjeće kod slojevite garderobe

6.1.3.2. Nedostaci modela za analizu sličnosti slika

Ako su pravilno detektovani granični okviri sljedeći korak je da se za njih kreiraju vektorske reprezentacije i da se pronađu najsličniji odjevni predmeti iz kataloga proizvoda. Ipak, i u ovom koraku postoje određeni problemi:

- Nemogućnost razumijevanja konteksta iz upita. Pošto je model obučen da pronalazi vizuelno slične slike, on nije u stanju da pravilno odredi sličnost odjevnih predmeta koji možda nisu toliko vizuelno slični, ali pripadaju zajedničkom kontekstu. Npr. ako je upit majica sa likom iz neke serije, a katalog proizvoda ima samo majice sa natpisima iz te serije, tada katalog vjerovatno neće odabrati te majice među najsličnijima, a važi i obrnuto. Takođe, ako je upit majica sa nekim likom specifičnog izgleda iz određene franšize, sistem će kao najsličnije da vrati majice sa likovima sličnog izgleda, a koji su možda iz potpuno drugih franšiza, a koje uopšte ne interesuju potrošača. Na slici 6.5 je ilustrovan dati problem, gdje majica sa likom čarobnjaka i majica sa engleskom riječi za čarobnjaka nisu uopšte vizuelno slične pa se ne bi pojavili u rezultatima pretrage jedna za drugu.

Slika 6.5: Primjer problema sa nerazumijevanjem konteksta kod pretrage majica u prodavnici

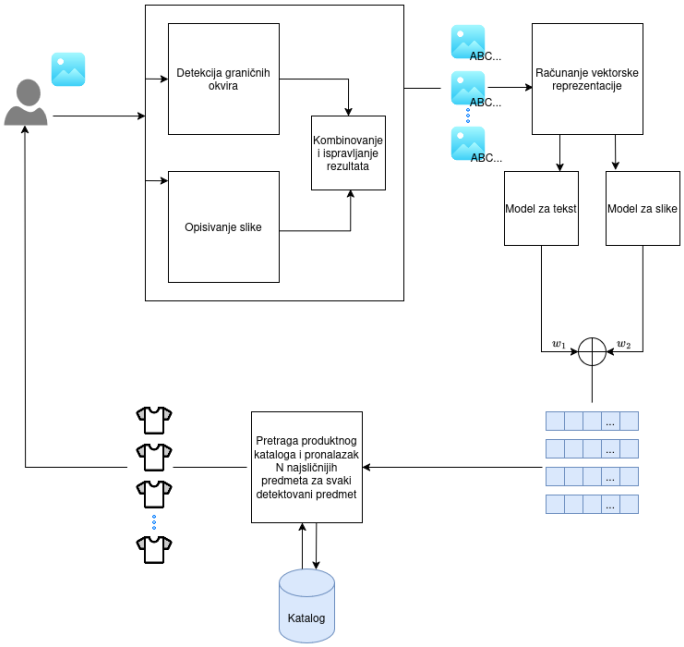
- Nedovoljna invarijantnost na različite načine prikaza nekih odjevnih predmeta. Kako bi se postigla invarijantnost u pretrazi prodavnice, potrebno je da se u trening skupu podataka nađe dovoljan broj uzoraka koji su prikazani pod različitim orijentacijama i u različitim okolnostima. Ipak model ima tendenciju da preferira vizuelno slične odjevne predmete i potpunu invarijantnost je nemoguće postići. Ako je upit košulja prikazana na nekoj osobi, a u katalogu proizvoda prodavača se nalazi ista ta košulja, ali na slici na kojoj je savijena, tada se ona vjerovatno neće naći među najsličnijim košuljama koje su vraćene kao rezultat pretrage, ova situacija je ilustrovana na slici 6.6.
- Poteškoće u pronalasku povezanih proizvoda. Pored sličnih odjevnih predmeta, korisnike ponekad zanima i pronalazak povezanih odjevnih predmeta. Na primjer ako korisnik kupuje duks neke određene robne marke, često želi da kupi i trenerku iste te robne marke, ili patike koje se po njemu dobro uklapaju sa tim duksom. Rješenje koje se bazira na analizi sličnosti slika će imati probleme u pronalasku povezanih proizvoda, jer nema mogućnost razumijevanja konteksta, niti veza između predmeta, a kako duks i trenerka npr. nisu naročito vizuelno slični, takvi predmeti se neće naći u preporukama jedni za druge[15].

[15] Ovaj problem nije analizirani u sklopu eksperimentalnog dijela rada, ali se može razmatrati kao jedan od pravaca daljnjeg istraživanja.

Slika 6.6: Primjer problema sa nerazumijevanjem konteksta i nedovoljne invarijantnosti kod pretrage košulja u prodavnici

6.2. PRIJEDLOG RJEŠENJA

Za rješavanje prethodno navedenih problema moguće je primijeniti opisivanje slika pomoću tehnika mašinskog učenja, pomoću kojeg se može simulirati razumijevanje konteksta. Na slici 6.7 je prikazana arhitekturalna šema jednog takvog sistema.



Slika 6.7: Šematski prikaz arhitekture sistema za pretragu prodavnice koja kombinuje tradicionalni pristup sa opisivanjem slika

Slično kao i kod sistema prikazanog na slici 6.2. prvi korak u sistemu uključuje detekciju graničnih okvira na ulaznoj slici I, kako bi se identifikovao skup $B = \{b_1, b_2, \dots, b_k\}$. Paralelno

s tim, koristi se model za generisanje opisa slike koji analizira ulaznu sliku i generiše skup tekstualnih opisa $O = \{o_1, o_2, \dots, o_l\}$, od kojih svaki opisuje jedan od detektovanih odjevnih predmeta. Sljedeći korak uključuje kombinovanje i ispravljanje rezultata, gdje se rješavaju prethodno opisani problemi. Problem generisanja lažno pozitivnih uzoraka rješava se tako što se, u slučaju kada model detektuje više okvira $|B| > 1$, generiše samo jedan opis $|O| = 1$, a zadržava samo najveći okvir $b_{\max}$, definisan površinom $w \cdot h$. Problem slojevite odjeće rješava se tako što se, u slučaju da model detektuje manje okvira $|B|$ nego što ima generisanih opisa $|O| > |B|$, opisi i okviri uparuju prema kategorijama (npr. "jakna", "majica"), dok se za opise koji ostanu neupareni generišu samo tekstualni vektori. Za svaki par (b_i, o_i), vektor slike v_i^s se računa pomoću CNN, dok se vektor opisa v_i^t generiše korišćenjem NLP modela kao što su Word2Vec, BERT ili sl. Rezultujući vektor v_i se dobija kao linearna kombinacija: $v_i = w_1 v_i^s + w_2 v_i^t$, gdje su w_1 i w_2 hiperparametri koji se određuju eksperimentalno. U slučaju da vektori v_i^s i v_i^t nisu iste dimenzije, vektor veće dimenzije se redukuje na dimenziju manjeg vektora. Nakon što su svi parovi (b_i, o_i) obrađeni, rezultujući vektori se upoređuju sa vektorskim reprezentacijama proizvoda iz kataloga $C = \{c_1, c_2, \dots, c_m\}$. Slični proizvodi se pronalaze na isti način kao i ranije.

Problem razumijevanja konteksta se rješava tako što generisani opisi sadrže informacije koje su od značaja za neki odjevni predmet, opisujući ujedno i neke kontekstualne informacije vezane za taj predmet, npr. naziv franšize i lika koji je naslikan na majici, ili da je riječ o savijenoj košulji. Dok bi se problem pronalaska povezanih proizvoda rješavao tako što bi se za sve kategorije odjevnih predmeta odredile druge kategorije odjevnih predmeta koje se uobičajeno zajedno kupuju. Potom bi se pretražili proizvodi iz povezanih kategorija i pronašli oni koji su najsličniji upitu i onda bi se prikazali korisniku.

7. EKSPERIMENTALNI DIO

U ovom poglavlju dat je opis eksperimentalnih rezultata za predloženo rješenje. Prvo je opisan proces obuke modela za detekciju graničnih okvira i analizu sličnosti slika. Nakon toga su opisani analizirani modeli za generisanje opisa slika i proces formiranja opisa iz sirovih podataka. Potom su dati rezultati evaluacije kvaliteta generisanih opisa koristeći standardne metrike opisane u 5.4. Konačno, analizirani su rezultati primjene predloženog rješenja na svim problemima navedenim u 6.1. Svi eksperimenti su pisani u Python programskom jeziku unutar *JupyterLab* okruženja.

7.1. Priprema modela za detekciju graničnih okvira i analizu sličnosti slika

Za potrebe obučavanja modela za detekciju objekata i modela za analizu sličnosti slika korišten je DF2 skup podataka. U pitanju je skup podataka koji sadrži veliki broj slika odjevnih predmeta iz trinaest kategorija, od kojih su neke uslikane od strane potrošača, a druge su profesionalne slike iz prodavnica odjeće. Na slici 7.1 su prikazane osnovne karakteristike DF2 trening skupa podataka, a slične su raspodjele i za validacioni i testni skup.

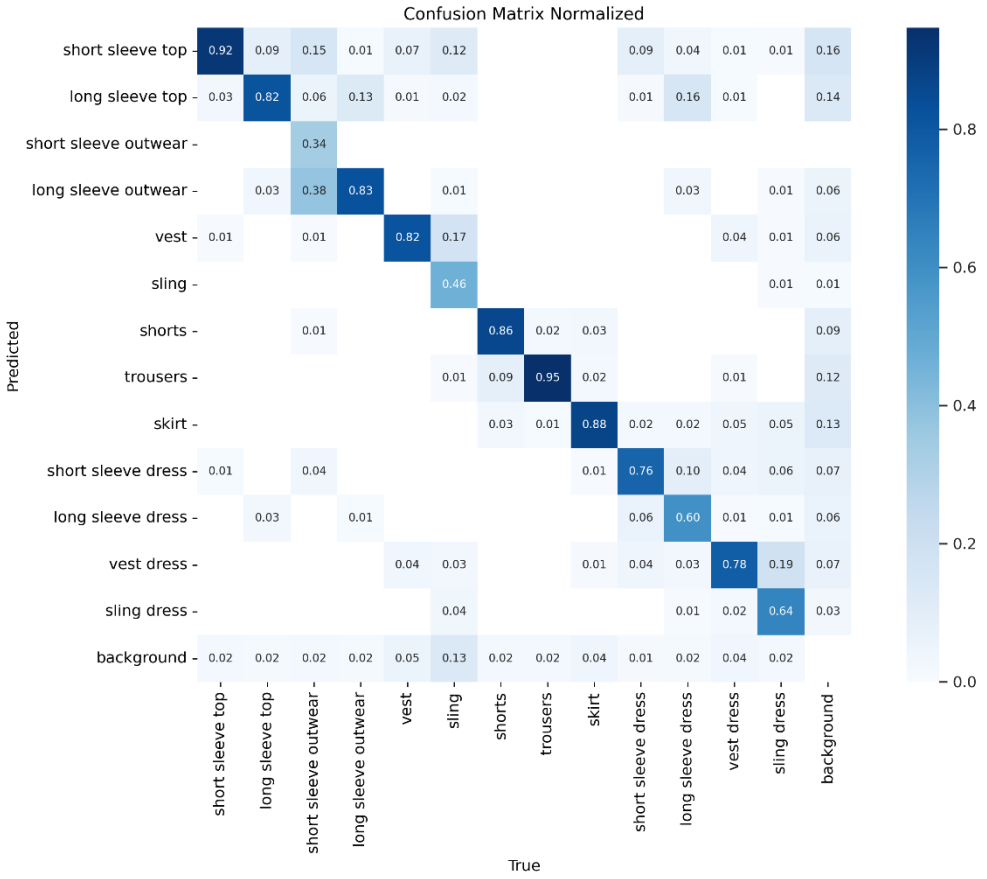
```
# Trening skup:
Ukupan broj slika: 191961
Ukupan broj odjevnih predmeta: 312186
Ukupan broj graničnih okvira: 312186
Broj graničnih okvira po kategoriji:
  long sleeve outwear: 13457
  trousers: 55387
  skirt: 30835
  short sleeve top: 71645
  vest dress: 17949
  shorts: 36616
  long sleeve top: 36064
  vest: 16095
  short sleeve dress: 17211
  sling dress: 6492
  long sleeve dress: 7907
  short sleeve outwear: 543
  sling: 1985
Ukupan broj parova prodavnica-potrošač: 14555
Ukupan broj odjevnih predmeta koji se ne mogu upariti: 94408
Raspodjela prema izvoru slike:
  potrošač: 59804
  prodavnica: 132157
```

Slika 7.1: Osnovne karakteristike DF2 trening skupa podataka

Za sve slike postoje precizno određeni granični okviri koji su iskorišteni prilikom obučavanja modela za detekciju objekata. Slike su uparene, tako da slike na kojima je prikazana ista garderoba iz prodavnice i one uslikane od strane potrošača imaju isti identifikator *pair_id*, pri čemu pojedinačni predmeti imaju svoj identifikator *style* kako bi se pravilno uparili, a oni predmeti koji ne mogu da se upare imaju *style* vrijednost 0.

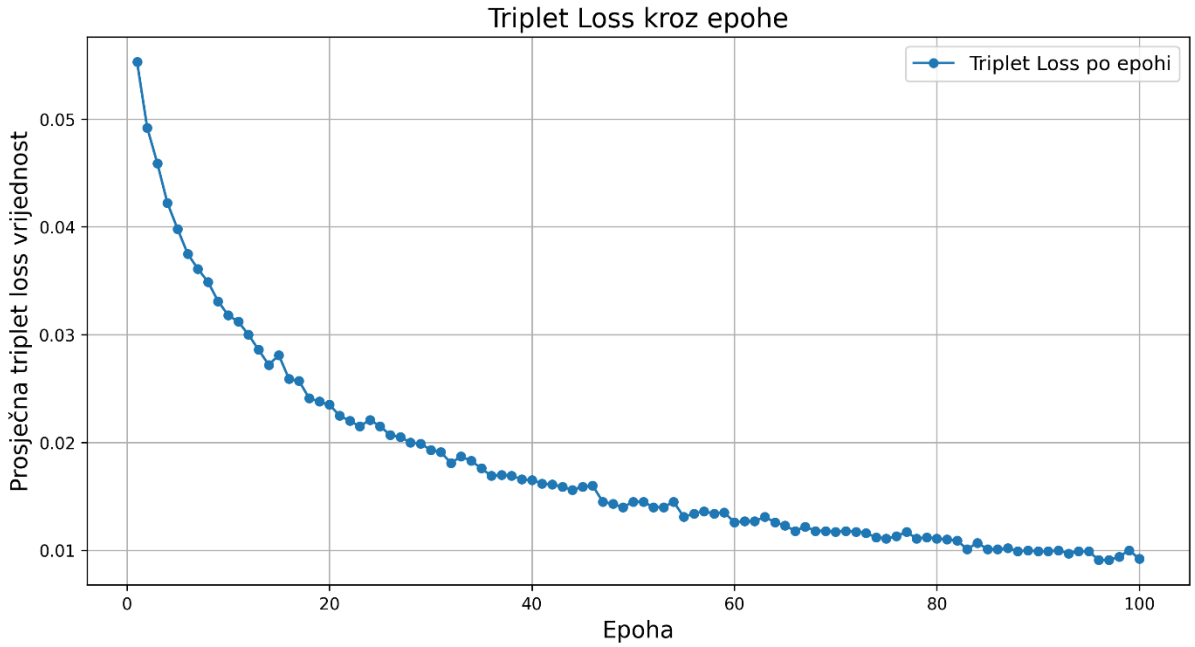
U prvom koraku su obučena tri modela za detekciju objekata na pomenutom skupu podataka. Korišteni su modeli YOLOv8 i YOLOv11, koji su dostupni u sklopu *ultralytics* Python biblioteke, te model Mask-RCNN koji je dostupan u sklopu *detectron2* biblioteke. Prilikom obuke kod sva tri modela je korišteno rano zaustavljanje, koje je opisano u drugom poglavlju, kako bi se izbjeglo preprilagođavanje trening podacima. Na slici 7.2 je data matrica konfuzije za DF2 validacioni skup za YOLOv11 model. Rezultati su najbolji za one kategorije

koje je teže vizuelno pomiješati i za koje je bilo najviše trening uzoraka, a gori su za one klase kod kojih i ljudi ponekad pogrešno dodijele labelu i gdje je bilo najmanje trening uzoraka.



Slika 7.2 Matrica konfuzije za YOLOv11 model

U narednom koraku je obučen model za analizu sličnosti slika, u pitanju je ResNet50 CNN dostupan u sklopu *torchvision* biblioteke. ResNet50 je prvobitno treniran na *ImageNet* skupu podataka. Ovaj pretrenirani model već posjeduje robusne karakteristike za ekstrakciju vizuelnih informacija, što ga čini idealnim za transfer tj. dotreniranje za zadatke poput analize sličnosti slika. Za potrebe dotreniranja, zamijenjen je originalni FFNN sloj sa kraja mreže, koji je generisao izlaz za 1.000 klasa, novim linearnim slojem koji generiše karakteristike u 128-dimenzionalni vektorski prostor. Dodatno, primijenjena je L2 normalizacija na izlazne vektore. Korišten je postupak obučavanje opisan u dijelu 6.1.2. Obučavanje se odvijalo kroz 100 epoha, u serijama (eng. *batches*) od 32 uzorka. Kao pozitivan uzorak je biran onaj koji ima istu kategoriju, *pair_id* i *style*, dok su svi ostali uzorci negativni, pri čemu je teže razlikovati one negativne uzorke koji su iz iste kategorije i imaju isti *pair_id* tj. kod kojih je samo drugačiji stil, zbog čega su ovakvi primjeri kad se nalaze unutar iste serije birani kao *hard negative*-i. U svakoj grupi je zastupljeno fiksno 16 različiti *pair_id* vrijednosti tj. klasa, a iz svake klase se nasumično biraju dva uzorka koji se mogu usidriti tj. da važi $style > 0$. Time se postiže balansiranost podataka unutar svake grupe, a u slučaju da nedostaje uzoraka koji se mogu usidriti, preostala mjesta se popunjavaju dodatnim negativnim primjerima. Na slici 7.3 je prikazan grafik prosječne *triplet loss* vrijednosti kroz trening epohe.



Slika 7.3 Triplet loss kroz epohe tokom obučavanja modela za analizu sličnosti slika

7.2. Analizirani modeli za generisanje opisa slika

U eksperimentalnom dijelu su analizirani modeli bazirani na transformatorskoj arhitekturi, kao i različiti multimodalni LLM-ovi, od kojih su neki komercijalno dostupni, a jedan od njih je besplatan i distribuiran pod licencom otvorenog koda. Prvobitna zamisao je bila i da se analiziraju i modeli bazirani na hibridnoj arhitekturi koja kombinuje CNN sa varijantama RNN, tj. sa LSTM ili GRU, ali su već prva ispitivanja pokazala da ovi modeli nisu dovoljno dobri za zadatak generisanje veoma specifičnih opisa kakvi su potrebni da bi se uspješno unaprijedili rezultati za pretragu prodavnice odjevnih predmeta, zbog čega su otpali za daljnja razmatranja.

Modeli sa transformatorskom arhitekturom za opisivanje slika pomoću teksta, koji su prethodili multimodalnim LLM, nisu bili direktno namijenjeni za detaljno opisivanje svih predmeta koji su prikazani na slici, već je fokus bio na generisanju uopštenog opisa za cjelokupnu sliku. Ovako generisani modeli nalaze svoju primjenu u drugim oblastima i za rješavanje drugih klasa problema, kao što je npr. automatsko generisanje alternativnog teksta i naslova slika na veb stranicama, što može biti od koristi osobama sa poteškoćama u vidu ili za pretragu multimedijalnog sadržaja npr. konceptualno sličnih slika koje bi se koristile kao pozadina za radnu površinu. Ipak, određeni modeli poput GIT, BLIP2 i Florence-2 su u stanju da generišu relativno detaljne opise, ali ni ti modeli ne mogu da generišu struktuiran opis, gdje se jasno izdvajaju pojedinačni predmeti.

Zbog toga su u radu stariji modeli isključivo analizirani, zajedno sa multimodalnim LLM, za problem razumijevanja konteksta gdje je kao upit data majica, odnosno gdje je majica glavni predmet za koji se opis generiše. U tabeli 7.1 su dati svi analizirani modeli sa odgovarajućim primjenama u radu.

Tabela 7.1. Analizirani modeli za generisanje opisa slika

Model	**Primjena u radu**	**Opis**
GIT_BASE		

Florence2-base	Razumijevanje konteksta za majice	Dotreniran za opisivanje majica i košulja
BLIP-2 Opt. 2.7B		
gpt-4o-mini-2024-07-18	Svi navedeni problemi	Pretreniran komercijalni model
gpt-4o-2024-08-06		
claude-3-5-sonnet-20241022		
claude-3-haiku-20240307		
llama3.2-vision-11b		Pretreniran javno dostupan model

7.2.1. Transformatorski modeli

Za potrebe dotreniranja transformatorskih modela iskorišten je skup podataka sa opisima odjevnih predmeta formiran iz *Eureka-Attr* kataloga proizvoda, čije su karakteristike prikazane u tabeli 7.2 [47].

Tabela 7.2. Raspodjela Eureka-Attr skupa podataka sa opisima slika za odjevne predmete

	Kategorija	**Broj odjevnih predmeta**
Trening	*t-shirts*	100.000
	shirts	10.000
Validacioni	*t-shirts*	20.000
	shirts	2.000

Na slici 7.4 su prikazani neki nasumično odabrani uzorci iz Eureka-Attr skupa podataka. Pored prikazanih kolona svaki uzorak (odjevni predmet) ima svoj jedinstveni identifikator i URL slike uzorka.

Name	Type	Brand	Collection	Gender	Color	Material	Sleeve Length	Pattern
Pointelle Knit Cuban-Collar Shirt	Shirt	Forever 21		Male	White	Cotton	Short sleeves	Plain
Men's Signature Denim Workshirt	Shirt	L.L.Bean		Male	Blue	Denim	Long sleeves	Plain
Polo Pony Striped Oxford Shirt in Blue	Shirt	Ralph Lauren		Male	Blue	Cotton	Long sleeves	Striped
Tokyo Address Long Sleeve Tee - White	T-shirt	Deus Ex Machina	Deus Classics	Unisex	White	Cotton	Long sleeves	Plain
Mock Two-Piece Short-Sleeve Collared Knit Top	T-shirt	Deepwood		Male	Brown	Cotton	Short sleeves	Striped
Oversized Organic Long Sleeve T-Shirt - Purple Haze	T-shirt	Colorful Standard		Unisex	Purple	Cotton	Long sleeves	
Cotton-jersey T-shirt with mirror-effect logo print	T-shirt	Boss		Male	White	Cotton	Short sleeves	
Hasta Muerte Widow Rose White T-Shirt	T-shirt	Zumiez		Unisex	White	Cotton	Short sleeves	Graphic
Loop Button Linen Shirt	Shirt	TOAST		Male	Slate	Linen	Full length	
Seattle Seahawks Air Essential Men's T-Shirt	T-shirt	Nike		Male	Blue	Cotton	Short sleeves	Graphic
Heavy Every Day Tee - Vintage Black	T-shirt	mnml		Male	Black	Cotton	Short sleeves	
8THWNDR Spider Cursor Black T-Shirt	T-shirt	Zumiez		Male	Black	Cotton	Short sleeves	Graphic
Cotton Blend Three Button Polo - Black/White	Shirt	Paul Fredrick		Male	Black	Cotton	Spandex	Short sleeves
Men's Athletic Eco Short Sleeve Tee	T-shirt	Mizuno		Male	Navy	Polyester	Short sleeves	
ASOS DESIGN knit polo with stripe pattern	Shirt	ASOS		Male	Navy	Cotton	Short sleeves	Striped
Deus Ex Machina Takoyaki Graphic T-Shirt	T-shirt	Deus Ex Machina		Male	Black	Cotton	Short sleeves	Graphic
Loose T-shirt with Balmain Signature embroidery	T-shirt	Balmain		Male	Black	Cotton	Short sleeves	Plain
Printed Boxy-Fit Short Sleeve T-Shirt	T-shirt	Bershka		Male	Beige	Cotton	Short sleeves	Graphic
Rick and Morty Graphic Tee	T-shirt	Licensed Character		Male	Black	Cotton	Short sleeves	
DGA Drama Black T-Shirt	T-shirt	Zumiez		Unisex	Black	Cotton	Short sleeves	Graphic
Seafoam 4s Flontae T-Shirt Be Good Graphic	T-shirt	Flontae Clothing		Unisex	Black	Cotton	Short sleeves	Graphic
Slim Secret Wash Cotton Poplin Shirt	Shirt	J.Crew		Male	Gray	Cotton	Long sleeves	Checked
Nike Mens Suns ES City Edition T-Shirt	T-shirt	Nike	City Edition	Male	Purple	Cotton	Short sleeves	Graphic

Slika 7.4 Nasumično odabrani uzorci iz Eureka-Attr skupa podataka

Da bi se dotrenirao model za generisanje opisa za odjevne predmete potrebno je uzorke iz datog skupa podataka iskoristiti za formiranje referentnih opisa. U nastavku je dat listing sa pseudo-kodom za formiranje opisa za uzorke iz datog skupa podataka.

```
def form_caption(row):

# Ulaz:
  # row: red iz TSV fajla koji sadrži vrijednosti za sve ili neke od kolona navedenih u
tabeli 7.2

# Izlaz:
  # Formiran opis za dati proizvod

# 1. Dohvaćanje i čišćenje vrijednosti pojedinačnih kolona
name = lowercase(trim(name))
brand = lowercase(trim(brand))
# ...
pattern = lowercase(trim(pattern))

# Niz kolona od kojih će se formirati opis
caption_parts = []

# 2. Dodavanje boje i pola na početak, ako nisu sadržane u nazivu proizvoda
if color and not name.contains(color):
  caption_parts.append(color)
if gender and not name.contains(gender):
  caption_parts.append(gender)

# 3. Opis uvijek sadrži naziv proizvoda
caption_parts.append(name)

# 4. Dodavanje preostalih kolona ako nisu dio naziva proizvoda
if category and not name.contains(category):
  caption_parts.append(category)

if brand and not name.contains(brand):
  caption_parts.append(f"from {brand}")

if collection and not name.contains(collection):
  caption_parts.append(f"part of {collection} collection")

if material and not name.contains(material):
  caption_parts.append(f"made of {material}")

if sleeve_length and not name.contains(sleeve_length):
  caption_parts.append(f"with {sleeve_length}")

# 5. Ako šara nije dio naziva i ako je vrijednost kolone šara "graphic"
# onda se piše "graphic print", inače je obična šara

if pattern and not name.contains(pattern):
  if pattern == "graphic":
    caption_parts.append("and graphic print")
  else:
    caption_parts.append(f"and {pattern} pattern")

# 6. Spajanje svih dijelova opisa u konačni opis i vraćanje rezultata
return (" ".join(caption_parts))
```

Listing 7.1 Pseudo-kod algoritma za formiranje opisa za uzorke

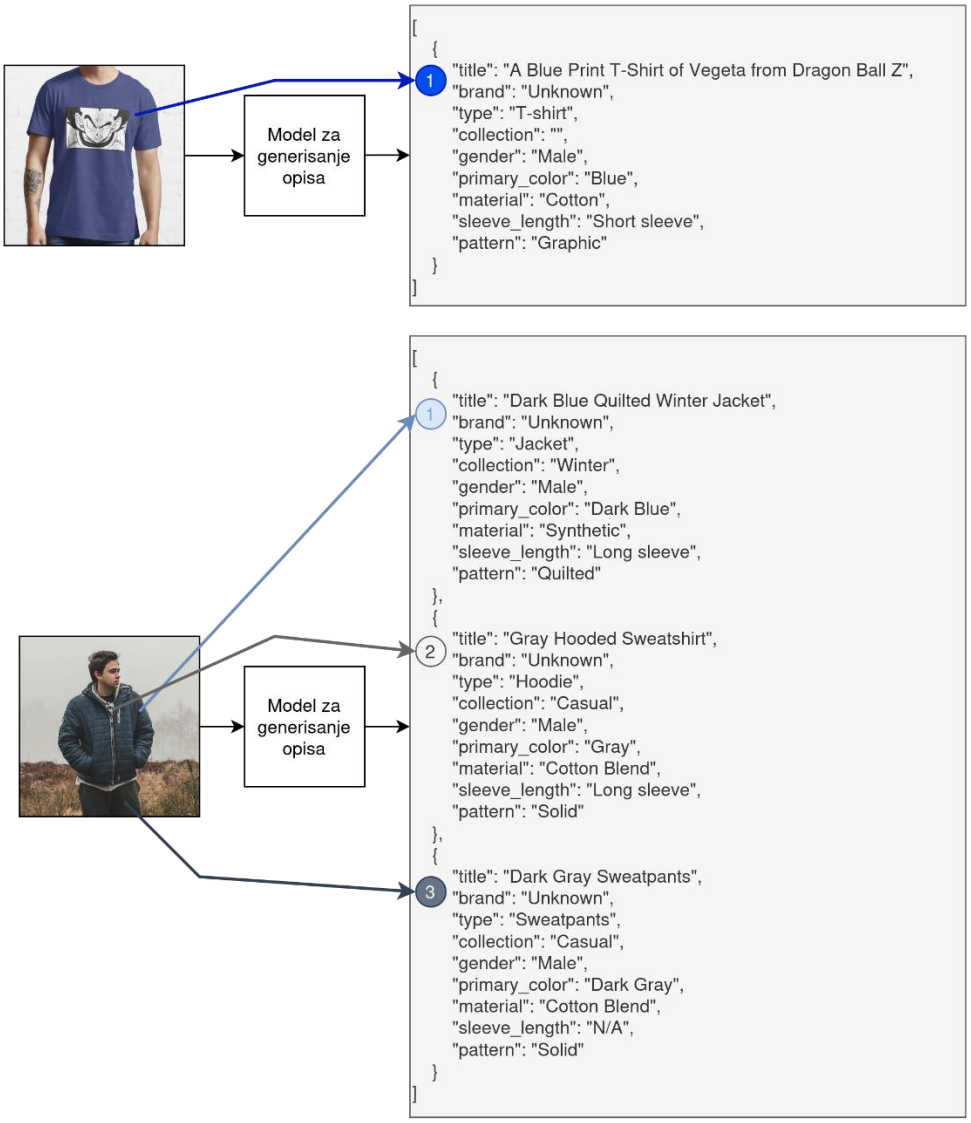
Kada se primjeni algoritam iz listinga 7.1 za uzorke sa slike 7.4 se dobiju opisi koji su dati u sklopu slike 7.5.

caption
white male pointelle knit cuban-collar shirt from forever 21 made of cotton with short sleeves and plain pattern
blue male men's signature denim workshirt from l.l.bean with long sleeves and plain pattern
blue male polo pony striped oxford shirt in blue from ralph lauren made of cotton with long sleeves
white unisex tokyo address long sleeve tee - white t-shirt from deus ex machina part of deus classics collection made of cotton with long sleeves and plain pattern
brown male mock two-piece short-sleeve collared knit top t-shirt from deepwood made of cotton with short sleeves and striped pattern
purple unisex oversized organic long sleeve t-shirt - purple haze from colorful standard made of cotton with long sleeves
white male cotton-jersey t-shirt with mirror-effect logo print from boss with short sleeves
white unisex hasta muerte widow rose white t-shirt from zumiez made of cotton with short sleeves and graphic print
slate male loop button linen shirt from toast with full length
blue male seattle seahawks air essential men's t-shirt from nike made of cotton with short sleeves and graphic print
black male heavy every day tee - vintage black t-shirt from mnml made of cotton with short sleeves
black male 8thwndr spider cursor black t-shirt from zumiez made of cotton with short sleeves and graphic print
black male cotton blend three button polo - black/white shirt from paul fredrick made of cotton, spandex with short sleeves
navy male men's athletic eco short sleeve tee t-shirt from mizuno made of polyester with short sleeves
navy male asos design knit polo with stripe pattern in navy and white shirt made of cotton with short sleeves and striped pattern
black male deus ex machina takoyaki graphic t-shirt in washed black made of cotton with short sleeves
black male loose t-shirt with balmain signature embroidery made of cotton with short sleeves and plain pattern
beige male printed boxy-fit short sleeve t-shirt from bershka made of cotton with short sleeves and graphic print
black male rick and morty graphic tee t-shirt from licensed character made of cotton with short sleeves
black unisex dga drama black t-shirt from zumiez made of cotton with short sleeves and graphic print
black unisex seafoam 4s flontae t-shirt be good graphic from flontae clothing made of cotton with short sleeves
gray male slim secret wash cotton poplin shirt from j.crew with long sleeves and checked pattern
purple male nike mens suns es city edition short sleeve logo t-shirt - orange/purple made of cotton with short sleeves and graphic print

Slika 7.5 Opisi formirani za uzorke sa slike 7.4

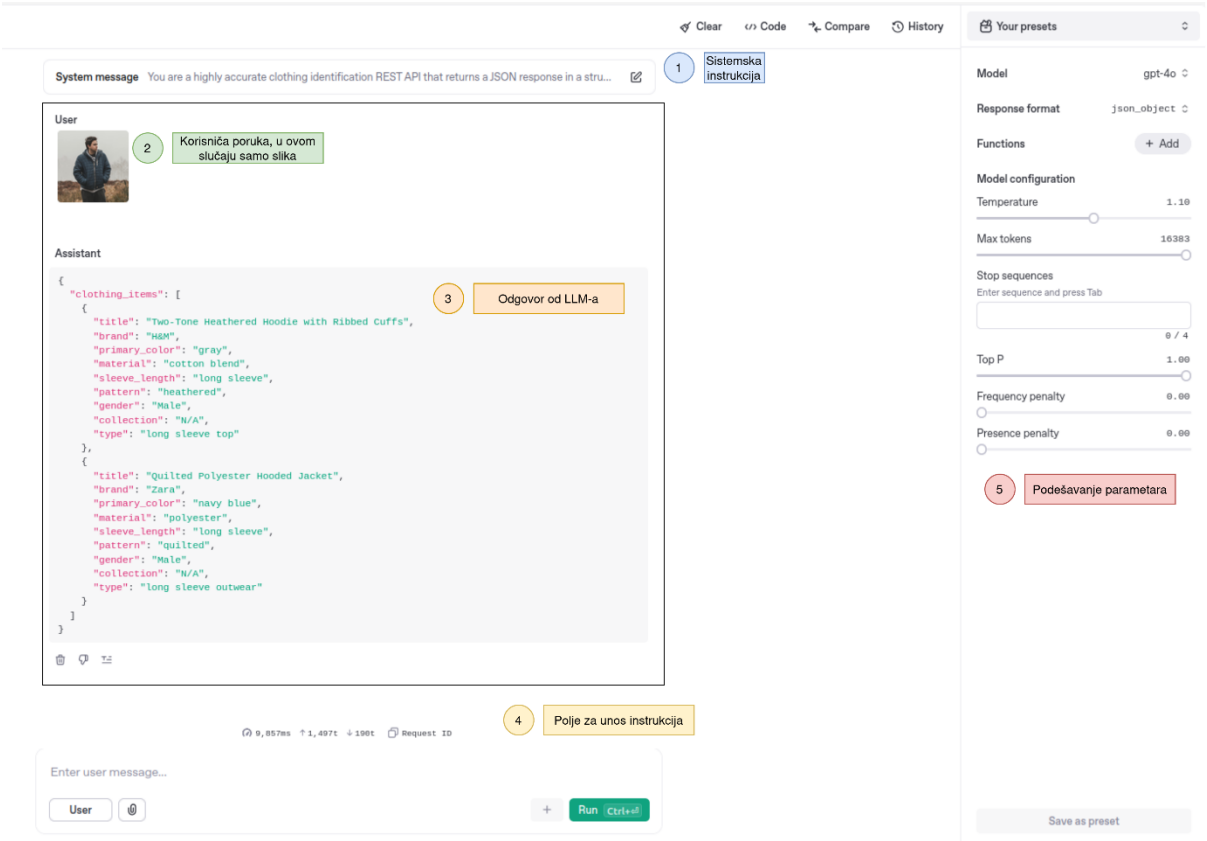
7.2.2. Multimodalni LLM-ovi

Primjenom multimodalnih LLM moguće je opisati sve predmete na slici, jer se kao odgovor može generisati struktuirani izlaz, npr. u JSON formatu [48-49]. Glavna prednost vraćanja rezultata u JSON formatu, umjesto direktno u tekstualnom obliku, ogleda se u tome što se dobija strukturirana i univerzalno prihvaćena forma podataka koju je znatno lakše parsirati i dalje obrađivati. Ovo omogućava jednostavnu integraciju sa bazama podataka, kao i automatsku obradu i filtriranje prema različitim kriterijumima, čime se značajno unaprjeđuje kvalitet i upotrebljivost kataloga proizvoda sa proširenim skupom informacija. Dodatno, ovakav pristup olakšava povezivanje tekstualnih opisa sa graničnim okvirima na slici, jer je informacija o kategoriji svakog objekta eksplicitno izdvojena kao poseban atribut u JSON strukturi. Suprotno tome, ukoliko bi se koristili isključivo tekstualni opisi, izdvajanje pojedinačnih atributa bilo bi znatno komplikovanije i podložno greškama, što otežava dalju automatizaciju i analizu. Na slici 7.6 su prikazani opisi koji su generisani primjenom ChatGPT-4o-mini modela. Prvi upit je majica koja sa sobom nosi dodatni kontekst, u ovom slučaju grafički print sa likom iz neke franšize iz popularne kulture, dok je drugi upit osoba koja nosi slojevitu garderobu, pri čemu LLM nije imao problema sa identifikovanje više slojeva garderobe, iako su neki od njih samo malim dijelom vidljivi. Kao i ranije, konačni opis za neki predmet se formira na osnovu algoritma iz listinga 7.1.



Slika 7.6 Opisi generisani pomoću multimodalnog LLM-a za dva upita

Kvalitet generisanog opisa uveliko zavisi od korištene instrukcije na ulazu LLM-a, zbog čega su za potrebe rada evaluirane različite instrukcije. Kao ispomoć pri *prompt engineering*-u korišteno je *OpenAI API - Chat Playground* okruženje koje je prikazano na slici 7.7. U pitanju je grafički interfejs sa određenim pogodnostima u radu sa *OpenAI API*-jem, gdje je moguće eksperimentisati sa različitim instrukcijama i odmah dobiti povratne informacije, čime se olakšava pronalazak kvalitetnijih instrukcija. Nakon niza inicijalnih eksperimenata, u razmatranje su uzete tri instrukcije koje su iskorištene za generisanje opisa za hiljadu proizvoda iz Eureka-Attr kataloga proizvoda, nakon čega su upoređene preko metrika za evaluaciju kvaliteta opisa.



Slika 7.7 OpenAI - Chat Playground

Najbolja instrukcija, koja je prikazana u sklopu listinga 7.2, je dalje korištena za sve LLM-ove[16]. Kategorije su namjerno ograničene tako da se koriste isti nazivi kao kod DF2 skupa podataka kako bi se kasnije opisi mogli upariti sa detektovanim graničnim okvirima odjevnih predmeta. Pored tih kategorija navedene su i neke druge koje nisu bile u sklopu DF2 skupa podataka, tako da ne može doći do konflikta. Iako se to rijetko dešava kada model dobije jasne instrukcije, postoji mogućnost da model halucinira i generiše i pogrešno imenuje neku kategoriju odjeće, pa se tada uparivanje ne može izvršiti. Ovaj problem se može riješiti podešavanjem temperature kao što je opisano u 4.1, ili tako što se analiziraju odgovori LLM modela koji se nisu mogli upariti, nakon toga se programski mapiraju najčešći propusti u prihvatljive nazive klasa. Alternativno primjenom *prompt engineering*-a potencijalno se može dobiti bolja instrukcija za koju će model manje halucinirati.

```
You are a highly accurate clothing identification REST API that returns a response in JSON
format. You are given a Base64 encoded image, and your job is to identify and name all
clothing items in it.

Request:
Base64 encoded image

Response:
{"items": [{"title": "...", "brand": "...", "collection": "...", "primary_color": "...",
"gender": "...", "type": "...", "material": "...", "sleeve_length": "...", "pattern": "..."
}, {"...": "..."}]}

Notes:
```

[16] Korištena instrukcija nije nužno najbolji izbor za svaki od analiziranih modela, jer svi oni rade pomalo drugačije. Ipak, pronalazak najadekvatnije instrukcije za svaki model je izvan okvira ovog rada i predstavlja jedan od mogućih pravaca daljnjeg istaživanja.

```
- The title should be very descriptive and include detailed information about the clothing
item
- If you can't name the brand of an item, you can pick the most likely brand.
- The brand can't have the value "Unknown".
- A collection is something like ["Nike Air Force 1", "adidas Samba", "adidas Superstar",
"Puma Suede", "New Balance 990", "Reebok Classic"]. Try to guess the collection whenever
possible.
- The names of the listed collection serve as examples, if you can't guess the collection
mark it as "Unknown"
- When identifying gender, choose from the following categories: ["Female", "Male",
Unisex"].
- You must also pick a type from the list ["short sleeve top", "long sleeve top", "short
sleeve outwear", "long sleeve outwear", "vest", "sling", "shorts", "trousers", "skirt",
"short sleeve dress", "long sleeve dress", "vest dress", "sling dress", "bags",
"accessories", "shoes", "hats", "scarves", "belts"]
```

Listing 7.2 Instrukcija koja je korištena da LLM izgeneriše struktuirane opise za garderobu

Kompanije *OpenAI* i *Anthropic* za rad sa svojim komercijalnim modelima nude plaćene REST API-je, gdje korisnici šalju zahtjeve i dobijaju odgovore u JSON formatu. Svaki zahtjev se naplaćuje, pri čemu su cijene određene od strane kompanije i podložne su promjenama[17]. Obe kompanije nude zvanične Python biblioteke koje pojednostavljuju rad sa njihovim REST API-jima. Nasuprot tome, Llama model nema sopstveni REST API interfejs, te se za rad s njim često koristi *Ollama*[18] okruženje pomoću kog se učita i pokrene Llama model. Ollama omogućava interakciju sa velikim brojem besplatnih LLM-ova putem konzolnog okruženja, Python biblioteke ili REST API-ija.

Različiti LLM-ovi imaju drugačiji pristup pri pisanju instrukcija, od korištenih *OpenAI ChatGPT* modeli i *Anthropic Claude* modeli podržavaju pisanje tzv. sistemskih poruka (eng. *system message*), što je samo olakšica da se razdvoji instrukcija koja je univerzalna i koja se šalje u sklopu svake interakcije, odnosno zahtjeva, npr. kao sistemska instrukcija je iskorištena instrukcija iz listinga 7.2. S druge strane Llama ne podržava sistemske poruke u kombinaciji sa slikama, pa se umjesto toga instrukcija iz listinga 7.2 šalje kao jedna od dvije obične poruke.

7.3. Rezultati evaluacije kvaliteta opisa

Za evaluaciju kvaliteta generisanih opisa slika su korištene metrike opisane u 5.4. Za BLEU metriku je iskorištena *sacrebleu* biblioteka, za ROUGE metriku je korištena *rouge_score* biblioteka, za METEOR metriku je korištena *nltk* biblioteka, za CIDER i SPICE metrike je korištena *pycocoevalcap* biblioteka, a za SPIDER je uzeta aritmetička sredina prethodne dvije metrike.

Za potrebe evaluacije generisanih opisa, detekcije slojevite odjeće, kao i za validiranje kvaliteta preporuka korištena su tri druga skupa podataka (disjunktni u odnosu na prethodne), a svi oni predstavljaju podskup *Eureka-PC* kataloga proizvoda [47]. Osnovne karakteristike ovih skupova podataka su date u tabeli 7.3.

Tabela 7.3 Raspodjela podskupa Eureka-PC kataloga proizvoda korištenog za analizu rješenja za četiri vrste opisanih problema

Kategorija	Broj odjevnih predmeta	Broj jednoznačnih predmeta	Problemi
t-shirts	100.000	80.870	Detekcija lažno pozitivnih graničnih okvira, (ne)razumijevanje konteksta
shirts	100.000	50.440	

[17] Prema trenutnoj šemi, tekst odnosno instrukcija koja se šalje u sklopu zahtjeva se tokenizuje, nakon čega se cijena odredi prema broju korištenih tokena iz zahtjeva i broju generisanih tokena u sklopu odgovora.
[18] https://ollama.com/

layered	10.000	0	Problem sa detekcijom slojevite odjeće

U tabeli 7.4 su dati rezultati evaluacije različitih modela korištenjem opisanih metrika, na skupu od 10.000 slika od čega je 5.000 iz kategorije *t-shirts* i 5.000 iz kategorije *shirts*, nasumično odabranih iz prethodno opisanog skupa.

Tabela 7.4 Rezultati evaluacije analiziranih modela na Eureka-Attr skupu podataka

Model	BLEU	METEOR	ROUGE-L F1	CIDEr	SPICE	SPIDEr
GIT_BASE	18,45	50,78	54,67	1,876	0,278	1,077
Florence2-base	20,12	53,34	57,23	2,101	0,301	1,201
BLIP-2 Opt. 2.7B	23,01	56,89	60,34	2,678	0,345	1,511
gpt-4o-mini-2024-07-18	34,57	78,84	79,19	4,997	0,578	2,788
gpt-4o-2024-08-06	36,91	81,23	81,34	5,123	0,593	2,858
claude-3-5-sonnet-20241022	33,12	76,45	77,56	4,678	0,541	2,609
claude-3-haiku-20240307	31,45	73,67	74,89	4,321	0,512	2,416
llama3.2-vision-11b	29,78	70,89	72,34	4,012	0,478	2,245

Najbolje rezultate ostvaruju *OpenAI ChatGPT* modeli, ali su dobri i rezultati ostvareni od strane *Anthropic Claude* modela i Llama modela. Može se činiti kako su dobijeni rezultati niži nego rezultati dobijeni u nekim drugim studijama, ali u pitanju je testiranje generisanih opisa na vrlo specifičnom domenu, pri čemu su svi referentni opisi detaljni, zbog čega su ocjene nešto niže nego kada bi se generisali uopšteniji opisi.

7.4. Analiza rezultata za predloženo rješenje

U nastavku su dati rezultati analize primjene predloženog rješenja na problemima koji su navedeni u poglavlju šest.

7.4.1. Problem detekcije lažno pozitivnih graničnih okvira

Rezultati evaluacije tri predložena modela za detekciju graničnih okvira i objekata, kao i pet predloženih LLM modela su dati u tabeli 7.5 Korišten je podskup proizvoda iz *Eureka-PC* kataloga, gdje je analizirano po pet hiljada proizvoda iz kategorije *t-shirts* i pet hiljada iz kategorije *shirts*, na svakoj slici je prikazan tačno jedan odjevni predmet, ali neki od odjevnih predmeta imaju na sebi naslikane ljudske likove.

Tabela 7.5 Rezultati evaluacije predloženih modela na problemu detekcije lažno pozitivnih uzoraka

Model	Prosječan broj detektovanih predmeta		Tačnost	
	t-shirts	*shirts*	*t-shirts*	*shirts*
gpt-4o-2024-08-06	1,003	1,002	99,60%	99,98%
gpt-4o-mini-2024-07-18	1,005	1,0076	99,58%	99,46%
claude-3-5-sonnet-20241022	1,003	1,0004	99,60%	99,60%
claude-3-haiku-20240307	1,007	1,0083	99,30%	99,30%
llama3.2-vision-11b	1,007	1,0013	99,30%	99,30%
YOLOv8	1,275	1,1712	77,90%	95,60%
YOLOv11	1,272	1,164	78,34%	95,60%
Mask R-CNN	1,28	1,1698	77,50%	77,50%

Najbolji rezultati su ostvareni korištenjem GPT-4o modela, ali i svi ostali predloženi modeli su ostvarili visoku tačnost na datom testnom skupu. U praktičnom sistemu naredni korak jeste usklađivanje izlaza oba modela. Na primjer, ako se koristi instrukcija iz listinga 7.2 u kombinaciji sa modelom za detekciju graničnih okvira obučenim na DF2 skupu podataka,

moguće je upariti odjevne predmete koji pripadaju istoj kategoriji. Međutim, prije samog uparivanja potrebno je eliminisati suvišne granične okvire. Jedan jednostavan, ali efikasan pristup jeste zadržavanje samo najvećeg graničnog okvira unutar svake kategorije. Alternativno, može se odabrati okvir kojem model dodjeljuje najveću vjerovatnoću pripadnosti određenoj kategoriji, ali kako se ponekad dešava da model daje veću sigurnost za lažno pozitivne uzorke – poput odjevnih predmeta prikazanih na likovima sa majice – sigurnija opcija ostaje filtriranje na osnovu veličine. Napredniji pristup uključuje korištenje modela poput CLIP-a [24], koji računa sličnost između opisa i isječaka slike dobijenih na osnovu graničnih okvira, čime se dodatno poboljšava tačnost uparivanja. Izbor optimalne metode zavisi od zahtjeva sistema: ako brzina odgovora nije kritična, upotreba CLIP-a može biti najsigurniji pristup, dok je, u scenarijima gdje je ključna brza obrada, jednostavnije filtriranje na osnovu veličine graničnog okvira efikasnija opcija.

7.4.2. Problem detekcije slojevite odjeće

Rezultati evaluacije predloženih modela na problemu detekcije slojevite odjeće su dati u tabeli 7.6 Korišten je podskup iz *Eureka-PC* kataloga proizvoda, od hiljadu proizvoda, slike proizvoda prikazuju ljude koji nose slojevitu garderobu, gdje je ručno određen tačan broj predmeta na svakoj slici iz trinaest kategorija koje su obrađene u DF2 skupu podataka. Dato ograničenje je uvedeno zbog poređenja sa modelima za detekciju koji su obučeni na DF2 skupu i mogu da detektuju samo navedene kategorije odjeće, dok u praksi LLM modeli mogu da detektuju i predmete iz drugih kategorija. Pravilna detekcija je slučaj kada neki model detektuje pravilan broj odjevnih predmeta, ali i pravilne kategorije za date predmete, dakle nije dovoljno pogoditi samo broj predmeta.

Tabela 7.6 Rezultati evaluacije predloženih modela na problemu detekcije slojevite odjeće

Model	**Tačnost**
gpt-4o-2024-08-06	70,80%
gpt-4o-mini-2024-07-18	71,00%
claude-3-5-sonnet-20241022	71,00%
claude-3-haiku-20240307	68,70%
llama3.2-vision-11b	67,10%
YOLOv8	35,40%
YOLOv11	36,40%
Mask R-CNN	41,40%

Najbolji rezultati su ostvareni pomoću GPT-4o modela, dok modeli za detekciju graničnih okvira znatno zaostaju za LLM modelima. Za razliku od prethodnog slučaja, gdje je bilo potrebno filtrirati višak detekcija, u ovom slučaju ne postoji mogućnost uparivanja svih generisanih opisa sa isječcima dobijenih na osnovu graničnih okvira zbog čega će neki od opisa ostati neupareni, pa će konačna vektorska reprezentacija za te predmete biti sačinjena samo od tekstualnog modaliteta, ali je to u svakom slučaju bolje nego da se ti predmeti u potpunosti zanemare, kao što je slučaj kod tradicionalnog pristupa.

7.4.3. Problem nerazumijevanja konteksta

Za potrebe analize razumijevanja konteksta kreiran je skup od 100 upita, pri čemu polovinu čine majice s printom, odnosno sa dodatnim kontekstom, a drugu polovinu obične majice bez printa. Skup je pažljivo izbalansiran kako bi rezultati analize bili validni. Naime, da je analiza vršena isključivo na upitima s print majicama, dobijeni rezultati bi bili preprilagođeni specifičnom problemu (eng. *overfitted*). S druge strane, za eksperiment je korišten katalog proizvoda od 10.000 majica, formiran na osnovu *Eureka-PC* kataloga proizvoda, gdje je za svaki upit određeno 20 najsličnijih majica. Poređeni su rezultati pretrage proizvoda u dva scenarija:

1. Korišćenje samo vizuelne reprezentacije slike.
2. Korišćenje linearna kombinacija vizuelne reprezentacije v_1 i tekstualne reprezentacije generisanog opisa v_2.

U tabeli su navedeni korišteni specifični modeli za sve vektorske reprezentacija teksta koje su opisane u petom poglavlju.

Model	Opis
Word2Vec[19]	Google-ov Word2Vec model3, koji je treniran na dijelu skupa podataka iz *Google News* korpusa, ukupno obuhvatajući oko 100 milijardi riječi. Model sadrži vektorske reprezentacije za oko 3 miliona jedinstvenih riječi i fraza, koje su predstavljene kao vektori dimenzije 300.
FastText[20]	FastText model treniran na skupu podataka *Common Crawl*, sa ukupno oko 600 milijardi riječi, koji sadrži dva miliona vektorskih reprezentacija riječi dimenzije 300.
GloVe[21]	GloVe model treniran na skupu podataka *Wikipedia* 2014 + *Gigaword* 5, koji sadrži ukupno oko šest milijardi riječi i 400 hiljada vektorskih reprezentacija riječi. Za potrebe rada korišćena je varijanta modela dimenzije 200.
ELMo[22]	ELMo model treniran na *Wikipedia* skupu podataka, koji sadrži ukupno oko pet milijardi riječi i vektore dimenzije 1024 [25].
BERT[23]	BERT model treniran na skupu podataka *BooksCorpus* i *English Wikipedia*, ukupno obuhvatajući oko 3,3 milijarde riječi. Korišćena je varijanta modela BERT-Base, koja generiše vektorske reprezentacije dimenzije 768.
text-embeddings-3-small[24]	Komercijalni model, vektora dimenzije 1536.

Eksperiment je uključivao poređenje dva pristupa pretrage po sličnosti: prvi pristup koristi samo vizuelne deskriptore dobijene primjenom YOLOv11+ResNet50 modela, dok drugi

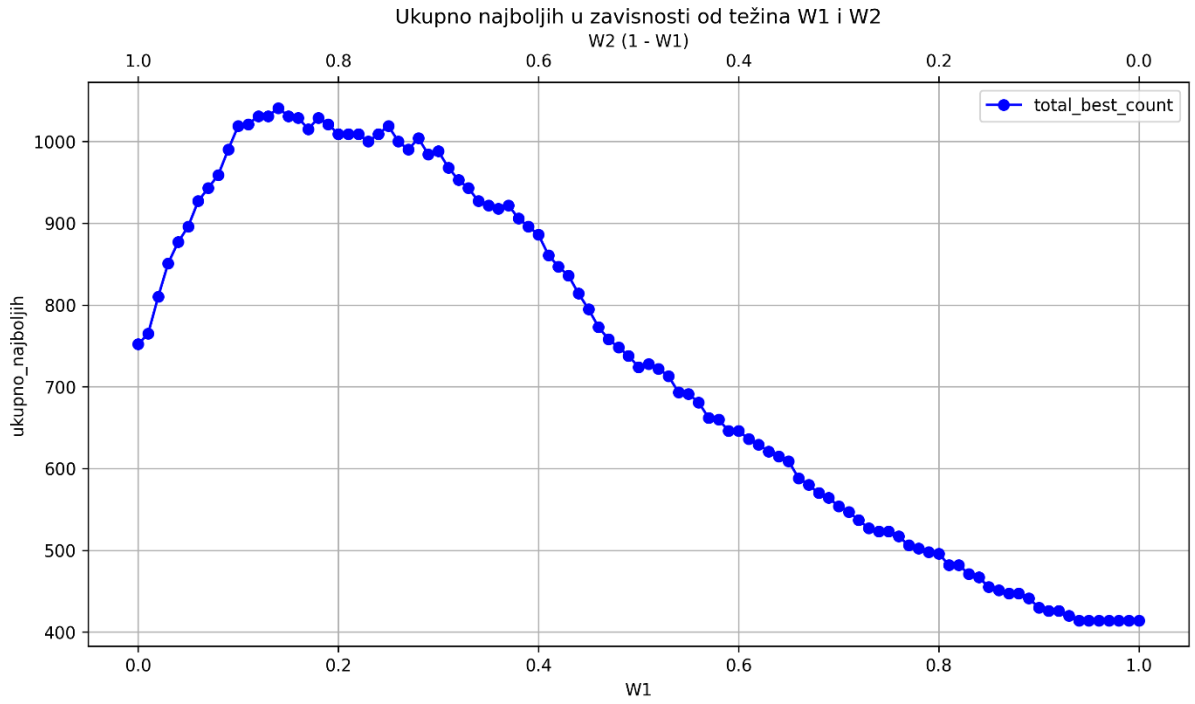
[19] https://code.google.com/archive/p/word2vec/
[20] https://fasttext.cc/docs/en/english-vectors.html
[21] https://nlp.stanford.edu/projects/glove/
[22] http://vectors.nlpl.eu/repository/
[23] https://huggingface.co/google-bert/bert-base-uncased
[24] https://platform.openai.com/docs/guides/embeddings/

kombinovani pristup koristi linearnu kombinaciju vizuelnih deskriptora i tekstualnih reprezentacija generisanih uz pomoć ChatGPT-4o LLM modela. Za svaki upit pronalazilo se dvadeset najsličnijih predmeta pomoću oba pristupa, a rezultati su upoređeni kako bi se utvrdilo koji model vraća više relevantnih predmeta. Ako su oba pristupa za dati upit pronašla isti broj relevantnih predmeta, smatrano je da su podjednako efikasni za taj upit. U slučaju da jedan pristup pronađe veći broj relevantnih predmeta od drugog (npr. vizuelni pronađe $14/20$, a kombinovani $15/20$), taj pristup se za taj upit smatrao boljim. Kako bi se postigla optimalna linearna kombinacija vektora v_1 i v_2, težine w_1 i w_2 su određene tako da zadovoljavaju uslov $w_1 + w_2 = 1$. Vrijednosti težina su ispitivane unutar intervala $[0,1]$ s korakom $\text{EPS} = 0{,}01$. Na taj način, za svaku kombinaciju težina testirani su rezultati kako bi se pronašao optimalan omjer. Eksperiment je pokazao da se bolji rezultati postižu korištenjem linearne kombinacije obje vektorske reprezentacije (v_1 i v_2) u odnosu na korištenje samo jedne od njih. Rezultati poređenja vizuelnog i kombinovanog pristupa za različite modele vektorske reprezentacije opisa prikazani su u tabeli 7.7.

Tabela 7.7 Rezultati evaluacije za problem razumijevanje konteksta nad majicama

Model	**% Bolji vizuelni pristup**	**% Isti rezultat**	**% Bolji kombinovani pristup**	**Optimalan omjer $w1 : w2$**
text-embedding-3-small	15	10	75	$14 : 86$
Word2Vec	12	26	62	$52 : 48$
FastText	9	27	64	$41 : 59$
GloVe	11	26	63	$38 : 62$
ELMo	8	27	65	$43 : 57$
BERT	14	16	70	$34 : 66$

Najbolje performanse postignute su s modelom *text-embedding-3-small*, pri čemu je optimalan omjer težina bio $w_1 : w_2 = 14 : 86$. Na slici 7.8 je dat grafik koji prikazuje ukupan broj tačno pronađenih najsličnijih odjevnih predmeta za sve upite za najbolji model. Za optimalan odnos težina, kombinovani pristup pronašao je ukupno 1041 najrelevantniji predmet za sve upite. Kada se koristi samo vizuelni pristup $w_2 = 1$, ukupan broj pronađenih najrelevantnijih predmeta iznosi 414. To znači da je kombinovanim pristupom pronađeno $\sim 151\%$ više relevantnih predmeta u poređenju s vizuelnim pristupom.

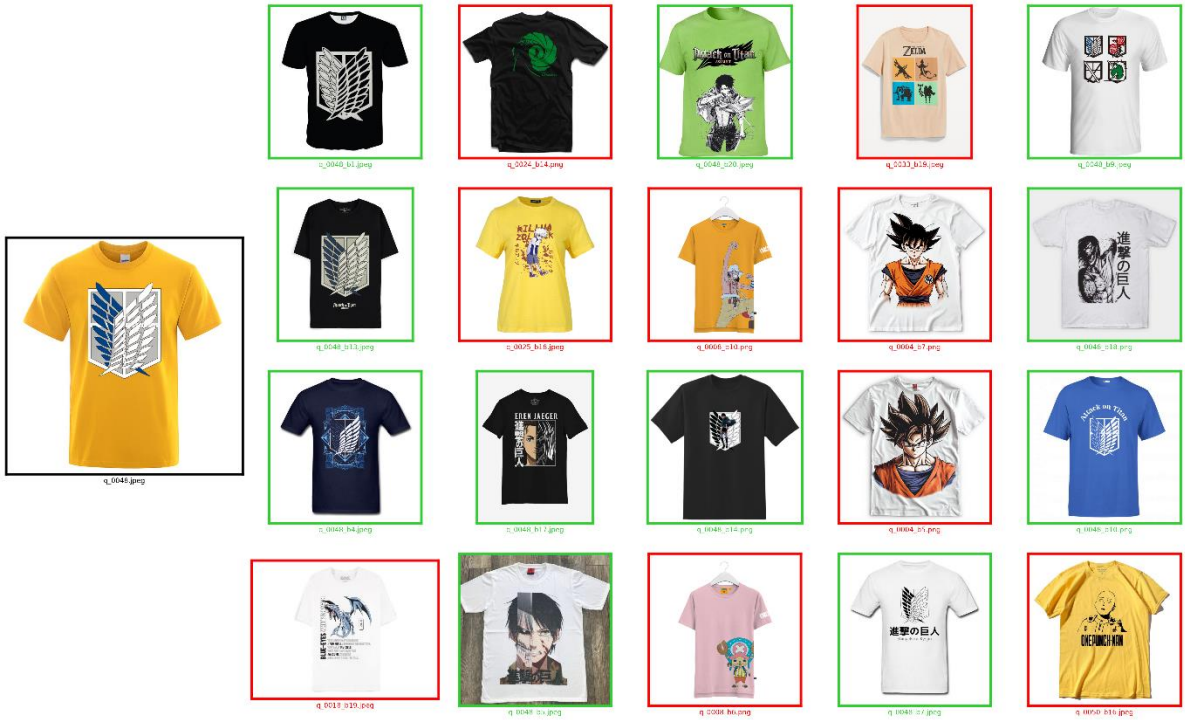


Slika 7.8 Ukupan broj tačno pronađenih najsličnijih odjevnih predmeta za sve upite za majice

Kao što ste vidi sa slike, sa porastom težine w_1, odnosno porastom značaja vizuelnog vektora opada broj pronađenih najsličnijih predmeta, jer opada moć modela da „razumije“ kontekst. Ipak, određeni udio vizuelnih informacija je potreban pošto je riječima teško opisati sve informacije sa slika. Na slikama 7.9 i 7.10, respektivno, su prikazani rezultati pretrage prodavnice korištenjem optimalne linearne kombinacije za *text-embedding-3-small* model. Odjevni predmeti koji su uokvireni zelenom bojom predstavljaju tačno pronađene najsličnije predmete iz kataloga, dok odjevni predmeti koji su zaokvireni crvenom bojom predstavljaju pogrešno pronađene odjevne predmete[25]. Za prvi upit je šest pogrešno pronađenih predmeta u rezultatima. U pitanju su slike koje prikazuju majice slične boje. Na tri od tih šest pogrešno pronađenih majica su prikazani likovi iz drugih popularnih animiranih serija, što je vjerovatno rezultat velike semantičke sličnosti, jer se o tim franšizama često priča u istom kontekstu, a i pored toga potiču iz iste zemlje. Slično je i za drugi prikazani upit, gdje je devet pogrešno pronađenih predmeta, koji mahom potiču iz drugih sličnih franšiza.

[25] Upiti su imenovani kao `q_0xzy`, dok su u katalogu najsličniji predmeti imenovani kao `q_0xyz_b#`. Pored tih najsličnijih predmeta za upite u katalogu proizvoda se nalaze i druge slike koje su označene sa `c_xzyw`. Ako u rezultatima za neki upit postoje slike koje imaju isti prefiks onda se one broje kao tačno pronađeni predmeti, u suprotnom su pogrešno pronađeni predmeti.

Slika 7.9 Rezultati pretrage prodavnice za majicu sa likom Pikachu-a iz animirane serije Pokemon[26]



Slika 7.10 Rezultati pretrage prodavnice za majicu sa grbom iz animirane serije Attack on Titan[27]

Konačno, analizirana je primjena na problemu razumijevanja konteksta i nedovoljne invarijantnosti kad su upiti košulje. Za potrebe analize, kreiran je skup od 50 upita, od čega polovinu čine savijene košulje, a drugu polovinu košulje na ljudima. Cilj je bio ispitati

[26] https://www.pokemon.com/us
[27] https://en.wikipedia.org/wiki/Attack_on_Titan

performanse različitih pristupa u pronalaženju sličnih predmeta unutar kataloga koji sadrži 10.000 predmeta, takođe formiranog na osnovu *Eureka-PC* kataloga proizvoda. Važno je napomenuti da ovaj katalog nije isti kao u prethodnom primjeru za majice, ali postoji određeno preklapanje kada su u pitanju nerelevantni predmeti. Za svaki upit identifikovano je deset najsličnijih predmeta. U analizi su upoređena dva pristupa:

1. Korišćenje samo vizuelne reprezentacije slike.
2. Korišćenje linearna kombinacija vizuelne reprezentacije v_1 i tekstualne reprezentacije generisanog opisa v_2.

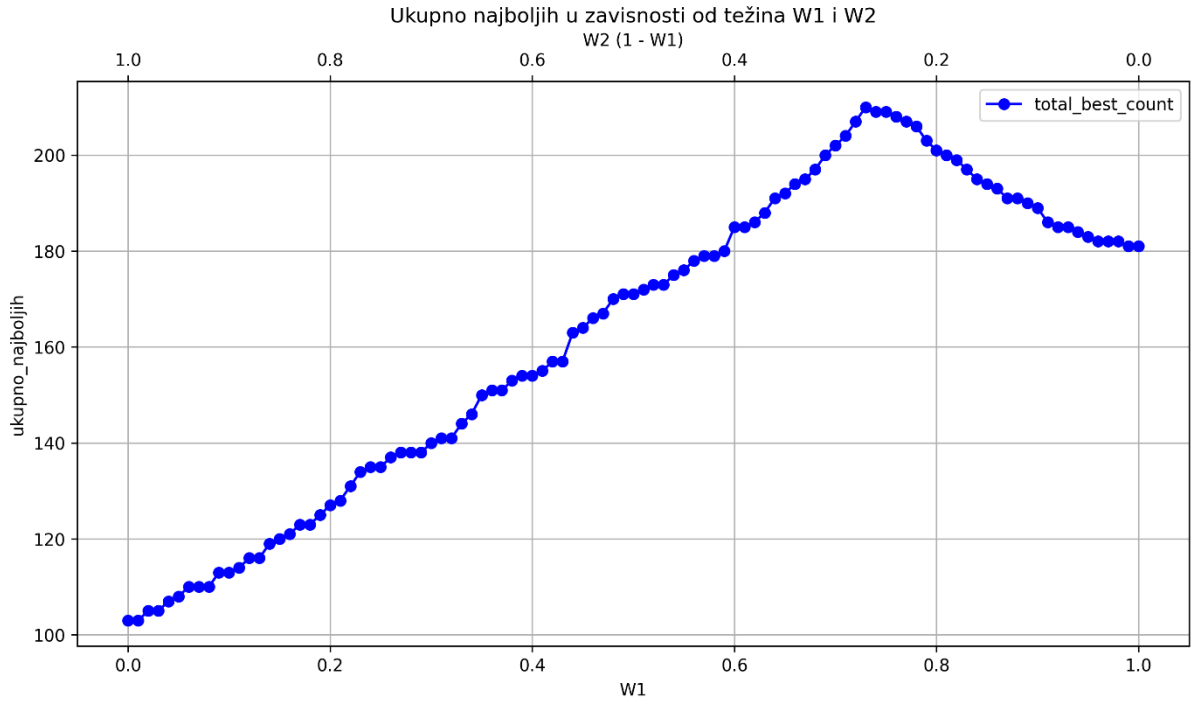
Testirani su sljedeći modeli: YOLOv11+ResNet50 i ChatGPT-4o. Jedan od izazova u analizi bio je nedostatak pouzdane detekcije graničnih okvira na savijenim košuljama prilikom upotrebe vizuelnih deskriptora. Uzrok ovog problema vjerovatno leži u nedostatku primjera sa savijenim košuljama u trening skupu podataka, zbog čega detekcija nije uvijek moguća, ili se loše odredi granični okvir. Iz tog razloga je, prilikom analize u slučaju da nije došlo do detekcije graničnog okvira, korištena cijela ulazna slika bez dodatnog isjecanja. Na ovaj način osigurano je da se ne donose zaključci isključivo na osnovu problema sa detekcijom graničnih okvira, već da se analizira i širi kontekst performansi modela. Kao i u prethodnom eksperimentu, za svaki upit analiziran je broj najrelevantnijih pronađenih predmeta kod oba pristupa, te je određeno koji pristup postiže bolje rezultate u pojedinačnim slučajevima. U situacijama kada oba pristupa pronalaze isti broj relevantnih predmeta, smatrano je da su jednako efikasni za dati upit, dok je pristup sa većim brojem pronađenih relevantnih predmeta ocijenjen kao bolji. Korišćenje linearne kombinacije vizuelne reprezentacije v_1 i reprezentacije generisanog teksta v_2 dalo je bolje rezultate u poređenju sa korišćenjem samo vizuelne reprezentacije. Optimalna linearna kombinacija težina w_1 i w_2 određena je tako da zadovoljava uslov $w_1 + w_2 = 1$, pri čemu su težine ispitivane u intervalu $[0,1]$ s korakom $\text{EPS} = 0{,}01$. U tabeli 7.8 su prikazani rezultati poređenja vizuelnog i kombinovanog pristupa za različite modele vektorske reprezentacije opisa.

Tabela 7.8 Rezultati evaluacije za problem razumijevanje konteksta i nedovoljne invarijantnosti nad košuljama

Model	**% Bolji vizuelni pristup**	**% Isti rezultat**	**% Bolji kombinovani pristup**	**Optimalan omjer $w1$: $w2$**
text-embedding-3-small	20	10	70	$73:27$
Word2Vec	16	26	58	$86:14$
FastText	12	26	62	$82:18$
GloVe	14	26	60	$83:17$
ELMo	14	24	62	$84:16$
BERT	14	22	64	$79:21$

Najbolji rezultati postignuti su s modelom *text-embedding-3-small*, pri čemu je optimalan omjer bio $w_1 : w_2 = 73 : 27$. Za optimalan odnos težina, kombinovani pristup pronašao je ukupno 210 najrelevantniji predmet za sve upite. Kada se koristi samo vizuelni pristup $w_2 = 1$, ukupan broj pronađenih najrelevantnijih predmeta iznosi 181. To znači da je kombinovanim pristupom pronađeno ~16% više relevantnih predmeta u poređenju s vizuelnim pristupom. Na slici 7.8 je dat grafik koji prikazuje ukupan broj tačno pronađenih najsličnijih odjevnih

predmeta za sve upite za najbolji model. Prema grafiku, model pokazuje bolje performanse kada je veći udio vizuelnih vektora. Dodatna analiza je pokazala sljedeći obrazac: Kod savijenih košulja, kombinacija vizuelne i tekstualne reprezentacije značajno je nadmašila pristupe bazirane samo na vizuelnim modelima. Ovo je očekivano jer tekstualni opisi omogućavaju bolji kontekstualni uvid u karakteristike savijenih košulja. Kod košulja koje su prikazane na osobama, vizuelni pristup (YOLOv11+ResNet50) pokazao se boljim u određenim slučajevima. To je zato što je šaru (eng. *pattern*) na košulji često teško precizno opisati riječima, dok vizuelni modeli poput ResNet50 mogu bolje prepoznati takve suptilne vizuelne detalje. Zbog toga je u ovim slučajevima prednost bila na strani vizuelne reprezentacije.



Slika 7.11 Ukupan broj tačno pronađenih najsličnijih odjevnih predmeta za sve upite za košulje

Na slikama A i B respektivno su prikazani rezultati pretrage prodavnice proizvoda za dvije košulje.

Slika 7.12 Rezultati pretrage prodavnice proizvoda za savijenu košulju

Slika 7.13 Rezultati pretrage prodavnice proizvoda za sivu košulju

Kod savijene košulje, među rezultatima pretrage često su se nalazile i druge savijene košulje koje po svim drugim karakteristikama nisu nužno bile slične upitnoj košulji, ali su bile prikazane u istoj "pozi". To sugeriše da sistem značajnu težinu pridaje položaju i obliku objekta na slici, odnosno da su deskriptori oblika i položaja imali dominantan uticaj na rangiranje rezultata. Slično, u drugom primjeru, među rezultatima su se mahom našle košulje istog kroja kao i upit, iako se razlikuju po boji. Utisak je da je sistem prepoznao i visoko rangirao predmete

na osnovu sličnosti oblika i dizajna, a ne samo na osnovu boje. Ovi primjeri jasno pokazuju širu problematiku pretrage po sličnosti, gdje različiti korisnici mogu različito vrjednovati pojedina obilježja kao što su oblik, boja ili tekstura. U praksi, korisnici bi mogli preferirati rezultate koji su sličniji po kroju, boji ili nekim drugim karakteristikama, u zavisnosti od svojih potreba, što je teško univerzalno predvidjeti ili opisati. Finija kontrola na osnovu korisničkih preferencija može se djelimično postići korištenjem predefinisanih opisa za određene varijante, uz mogućnost da korisnik sam bira kriterijume pretrage prema svojim preferencijama.

8. ZAKLJUČAK

U ovom radu opisan je prijedlog rješenja za primjenu opisa generisanih pomoću tehnika mašinskog učenja za problem pretrage proizvoda. Ovim radom pokazano je da je za uspješnu pretragu bolje kombinovati model za analizu sličnosti slika zajedno sa modelom za opisivanje slika, odnosno linearno kombinovanje vektorske reprezentacije dobijene na osnovu vizuelnih karakteristika sa vektorskom reprezentacijom generisanog opisa. Posebna pažnja je posvećena problemima sa klasičnim pristupom za pretragu proizvoda koji se oslanja samo na analizu sličnosti slika, gdje problemi potiču ili od modela za detekciju koji se nalazi na početku takvog sistema, ili od modela za analizu sličnosti koji se nalazi na kraju tog sistema. Na kraju je data analiza kako se ovi problemi uspješno rješavaju korištenjem modela za generisanje opisa. Mogući pravci daljeg istraživanja obuhvataju dotreniranje LLM-ova za dati specifični problem generisanja opisa za pretragu proizvoda, kao i primjena temeljnog *prompt engineering*-a u cilju dobijanja što boljih opisa koji bi detaljno opisali što veći broj karakteristike odjevnih predmeta, kao i testiranje sve većeg broja dostupnih LLM-ova. Budući da otvoreni multimodalni LLM modeli često omogućavaju direktno dobijanje vizuelnih vektorskih reprezentacija slike, jedan od potencijalnih pravaca daljeg istraživanja jeste korišćenje tih reprezentacija za pretragu proizvoda, čime bi se izbjegla potreba za generisanjem tekstualnog opisa. Pored toga, jedan pravac daljeg istraživanja odnosi se na problem pretrage i preporuke povezanih proizvoda. Na primjer, ako je korisnik zainteresovan za određeni gornji dio odjeće, poželjno je da mu se predloži i odgovarajući donji dio koji se stilski i funkcionalno uklapa sa prvim proizvodom. Ovakve veze i preporuke često nije moguće u potpunosti ostvariti samo analizom vizuelne sličnosti, dok bi analiza sličnosti opisa mogla omogućiti otkrivanje proizvoda koji se međusobno dopunjuju, na osnovu zajedničkih ili kompatibilnih karakteristika. Dodatno, generisani opisi mogu se koristiti i za kvalitetniju kategorizaciju i razvrstavanje kataloga proizvoda, na primjer prema materijalima, dužini, okolnostima korišćenja, sezonalnosti, trendu kojem pripadaju i drugim relevantnim osobinama. Ostvareni rezultati i prikazana metodologija potvrđuju da primjena savremenih metoda obrade vizuelnih i tekstualnih podataka može unaprijediti pretragu proizvoda, čime se otvaraju nove mogućnosti za razvoj još efikasnijih sistema u realnim aplikacijama.

LITERATURA

[1] D.-C. Pahonțu и E.-Ștefania Enache, „An Overview of AI-driven Recommendation Systems: Enhancing Personalization & User Experience (Qualitative Study)“, Student Thinkers and Advanced Research, tom 3, izd. 2, Art. izd. 2, Nov. 2024. Available at: https://opacj.org/star/article/view/539.

[2] D. G. Balasubramanian, „THE ROLE OF AI IN ENHANCING PERSONALIZATION IN ECOMMERCE: A STUDY ON CUSTOMER ENGAGEMENT AND SATISFACTION“, *Asian Pac. Econ. Rev.*, tom 17, izd. 2, Art. izd. 2, Nov. 2024.

[3] Orcun Sarioguz and Evin Miser, “Assessing the role of artificial intelligence in enhancing customer personalization: A study of ethical and privacy implications in digital marketing,” International Journal of Science and Research Archive, vol. 13, no. 2, pp. 812–825, Nov. 2024. doi:10.30574/ijsra.2024.13.2.2207.

[4] Y. Ge, R. Zhang, L. Wu, X. Wang, X. Tang, и P. Luo, „DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images“, 23. Januar 2019., *arXiv*: arXiv:1901.07973. doi: 10.48550/arXiv.1901.07973.

[5] D. Morelli, M. Cornia, and R. Cucchiara, "FashionSearch++: Improving Consumer-to-Shop Clothes Retrieval with Hard Negatives," in *Proceedings of the Italian Information Retrieval Workshop (IIR)*, 2021. [Na Internetu]. Available at: https://api.semanticscholar.org/CorpusID:237621688.

[6] P. Alirezazadeh, F. Dornaika, и A. Moujahid, „Deep Learning with Discriminative Margin Loss for Cross-Domain Consumer-to-Shop Clothes Retrieval“, *Sensors*, tom 22, izd. 7, Art. izd. 7, Jan. 2022, doi: 10.3390/s22072660.

[7] P. Norvig и S. Russell, *Artificial Intelligence: A Modern Approach, Global Edition*, 4th edition. Harlow: Pearson, 2021.

[8] I. Goodfellow, Y. Bengio, и A. Courville, *Deep Learning*, Illustrated edition. Cambridge, Massachusetts: The MIT Press, 2016.

[9] L. G. Serrano и S. Thrun, *Grokking machine learning*. Shelter Island: Manning, 2021.

[10] V. Risojević, *Multimedijalni sistemi*. Banja Luka: Univerzitet u Banjoj Luci, Elektrotehnički fakultet, 2018.

[11] A. Glassner, *Deep Learning: A Visual Approach*, Illustrated edition. San Francisco, CA: No Starch Press, 2021.

[12] „Normalization in Machine Learning: A Breakdown in detail“, OpenGenus IQ: Computing Expertise & Legacy. Приступљено: 20. Novembar 2023. [Na Internetu]. Available at: https://iq.opengenus.org/normalization-in-detail/

[13] V. R. Joseph, „Optimal Ratio for Data Splitting“, *Stat. Anal. Data Min. ASA Data Sci. J.*, tom 15, izd. 4, str. 531–538, Avg. 2022, doi: 10.1002/sam.11583.

[14] X. Ying, „An Overview of Overfitting and its Solutions“, *J. Phys. Conf. Ser.*, tom 1168, izd. 2, str. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[15] H. Gu, Y. Wang, S. Hong, и G. Gui, „Blind Channel Identification Aided Generalized Automatic Modulation Recognition Based on Deep Learning“, *IEEE Access*, tom PP, str. 1–1, Avg. 2019, doi: 10.1109/ACCESS.2019.2934354.

[16] V. Bui, N. T. Le, V. H. Nguyen, J. Kim, и Y. M. Jang, „Multi-Behavior with Bottleneck Features LSTM for Load Forecasting in Building Energy Management System“, *Electronics*, tom 10, izd. 9, str. 1026, Apr. 2021, doi: 10.3390/electronics10091026.

[17] L. Xiong *i ostali*, „DCAST: A Spatiotemporal Model with DenseNet and GRU Based on Attention Mechanism“, *Math. Probl. Eng.*, tom 2021, str. 1–12, Feb. 2021, doi: 10.1155/2021/8867776.

[18] „Transformer: A Novel Neural Network Architecture for Language Understanding“. Pristupljeno: 19. Novembar 2024. [Na Internetu]. Available at: http://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/
[19] ketan, „Transformers Explained Visually - Overview of Functionality“, Ketan Doshi Blog. Pristupljeno: 16. Februar 2025. [Na Internetu]. Available at: https://ketanhdoshi.github.io/Transformers-Overview/
[20] A. Vaswani *i ostali*, „Attention Is All You Need“, 02. Avgust 2023., *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
[21] J. Wang *i ostali*, „GIT: A Generative Image-to-text Transformer for Vision and Language“, 15. Decembar 2022., *arXiv*: arXiv:2205.14100. doi: 10.48550/arXiv.2205.14100.
[22] ketan, „Transformers Explained Visually - How it works, step-by-step“, Ketan Doshi Blog. Pristupljeno: 03. Februar 2025. [Na Internetu]. Available at: https://ketanhdoshi.github.io/Transformers-Arch/
[23] Grant Sanderson, *Visualizing transformers and attention | Talk for TNG Big Tech Day '24*, (2024.). Pristupljeno: 16. Februar 2025. [Na Internetu Video]. Available at: https://www.youtube.com/watch?v=KJtZARuO3JY
[24] ketan, „Transformers Explained Visually - Multi-head Attention, deep dive“, Ketan Doshi Blog. Pristupljeno: 16. Februar 2025. [Na Internetu]. Available at: https://ketanhdoshi.github.io/Transformers-Attention/
[25] S. Raschka, Build a Large Language Model (from Scratch). Shelter Island, NY: Manning Publications, 2025.
[26] S. R. PhD, „Understanding Multimodal LLMs“. Pristupljeno: 23. Decembar 2024. [Na Internetu]. Available at: https://magazine.sebastianraschka.com/p/understanding-multimodal-llms
[27] J. Eisenstein, Introduction to Natural Language Processing. Cambridge, MA: The MIT Press, 2019.
[28] T. Mikolov, K. Chen, G. Corrado, i J. Dean, „Efficient Estimation of Word Representations in Vector Space“, 07. Septembar 2013., *arXiv*: arXiv:1301.3781. Pristupljeno: 04. Novembar 2024. [Na Internetu]. Available at: http://arxiv.org/abs/1301.3781
[29] A. Joulin, E. Grave, P. Bojanowski, i T. Mikolov, „Bag of Tricks for Efficient Text Classification“, 09. Avgust 2016., *arXiv*: arXiv:1607.01759. Pristupljeno: 11. Novembar 2024. [Na Internetnu]. Available at: http://arxiv.org/abs/1607.01759
[30] A. CR, „Word Embeddings in NLP | Word2Vec | GloVe | fastText“, Analytics Vidhya. Pristupljeno: 04. Novembar 2024. [Na Internetu]. Available at: https://medium.com/analytics-vidhya/word-embeddings-in-nlp-word2vec-glove-fasttext-24d4d4286a73
[31] J. Pennington, R. Socher, i C. Manning, „GloVe: Global Vectors for Word Representation“, u *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, и W. Daelemans, Ur., Doha, Qatar: Association for Computational Linguistics, Okt. 2014, str. 1532–1543. doi: 10.3115/v1/D14-1162.
[32] M. E. Peters *i ostali*, „Deep contextualized word representations“, 22. Mart 2018., *arXiv*: arXiv:1802.05365. doi: 10.48550/arXiv.1802.05365.
[33] J. Devlin, M.-W. Chang, K. Lee, i K. Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“, 24. Maj 2019., *arXiv*: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[34] Umar Jamil, *BERT explained: Training, Inference, BERT vs GPT/LLamA, Fine tuning, [CLS] token*, (2023.). Pristupljeno: 21. Januar 2025. [Na Internetu Video]. Available at: https://www.youtube.com/watch?v=90mGPxR2GgY
[35] K. Papineni, S. Roukos, T. Ward, и W.-J. Zhu, „Bleu: a Method for Automatic Evaluation of Machine Translation“, u *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, i D. Lin, Ur., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Juli 2002, str. 311–318. doi: 10.3115/1073083.1073135.
[36] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, i E. Erdem, „Re-evaluating Automatic Metrics for Image Captioning“, u *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, M. Lapata, P. Blunsom, i A. Koller, Ur., Valencia, Spain: Association for Computational Linguistics, Апр. 2017, str. 199–209. Pristupljeno: 13. Januar 2025. [Na Internetu]. Available at: https://aclanthology.org/E17-1019/
[37] S. Castilho, S. Doherty, F. Gaspari, i J. Moorkens, „Approaches to Human and Machine Translation Quality Assessment: From Principles to Practice“, 2018, str. 9–38. doi: 10.1007/978-3-319-91241-7_2.
[38] S. Banerjee i A. Lavie, „METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments“, u *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, J. Goldstein, A. Lavie, C.-Y. Lin, i C. Voss, Ur., Ann Arbor, Michigan: Association for Computational Linguistics, Juni 2005, str. 65–72. Pristupljeno: 16. Juni 2024. [Na Internetu]. Available at: https://aclanthology.org/W05-0909
[39] C.-Y. Lin, „ROUGE: A Package for Automatic Evaluation of Summaries“, u *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Juli 2004, str. 74–81. Pristupljeno: 09. Juni 2024. [Na Internetu]. Available at: https://aclanthology.org/W04-1013
[40] A. de S. Inácio i H. S. Lopes, „Evaluation metrics for video captioning: A survey“, *Mach. Learn. Appl.*, tom 13, str. 100488, Sep. 2023, doi: 10.1016/j.mlwa.2023.100488.
[41] R. Vedantam, C. L. Zitnick, i D. Parikh, „CIDEr: Consensus-based Image Description Evaluation“, 03. Juni 2015., *arXiv*: arXiv:1411.5726. doi: 10.48550/arXiv.1411.5726.
[42] P. Anderson, B. Fernando, M. Johnson, i S. Gould, „SPICE: Semantic Propositional Image Caption Evaluation“, 29. Juli 2016., *arXiv*: arXiv:1607.08822. doi: 10.48550/arXiv.1607.08822.
[43] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, i K. Murphy, „Improved Image Captioning via Policy Gradient optimization of SPIDEr“, u *2017 IEEE International Conference on Computer Vision (ICCV)*, Okt. 2017, str. 873–881. doi: 10.1109/ICCV.2017.100.
[44] J. Redmon, S. Divvala, R. Girshick, i A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“, 09. Maj 2016., *arXiv*: arXiv:1506.02640. doi: 10.48550/arXiv.1506.02640.
[45] K. He, G. Gkioxari, P. Dollár, i R. Girshick, „Mask R-CNN“, 24. Januar 2018., *arXiv*: arXiv:1703.06870. doi: 10.48550/arXiv.1703.06870.
[46] D. Hoiem, Y. Chodpathumwan, i Q. Dai, „Diagnosing error in object detectors“, у *Computer vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, и C. Schmid, Ур., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, стр. 340–353.
[47] mscloading888, mscloading888/MSC-AM-Datasets. (05. maj 2025.). Pristupljeno: 05. maj 2025. [Na internetu]. Dostupno na: https://github.com/mscloading888/MSC-AM-Datasets

[48] „Introducing Structured Outputs in the API“. Pristupljeno: 12. Januar 2025. [Na Internetu]. Available at: https://openai.com/index/introducing-structured-outputs-in-the-api/
[49] „Increase output consistency (JSON mode)“, Anthropic. Pristupljeno: 12. Januar 2025. [Na Internetu]. Available at: https://docs.anthropic.com/en/docs/test-and-evaluate/strengthen-guardrails/increase-consistency

Biografija autora

Aleksije Mićić, dipl. inž. elektrotehnike, rođen 06.03.1997. godine u Novom Sadu. Osnovnu i srednju školu je završio u Mrkonjić Gradu, kao odličan đak. Nakon toga je pohađao Elektrotehnički fakultet Univerziteta u Banjoj Luci, gdje se upisao na studijski program računarstva i informatike. Osnovne studije je završio u junu 2020. godine, odbranom rada pod nazivom "Razvoj Android aplikacija korištenjem programskog jezika Kotlin" i sa prosječnom ocjenom 8,49. Od oktobra 2019. godine je zaposlen u kompaniji *Bravo Systems d.o.o.* Banja Luka, gdje radi kao softverski inženjer. U okviru svog angažmana učestvuje na raznovrsnim projektima iz oblasti mašinskog učenja, digitalnog marketinga i razvoju sistema za preporuke.

Прилог 2.

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ПОДАЦИ О АУТОРУ ОДБРАЊЕНОГ МАСТЕР/МАГИСТАРСКОГ РАДА

Име и презиме аутора мастер/магистарског рада: **Алексије Мићић**

Датум, мјесто и држава рођења аутора: **06.03.1997, Нови Сад, Србија**

Назив завршеног факултета/Академије аутора и година дипломирања:

Електротехнички факултет, Универзитет у Бањој Луци, 2020. година

Датум одбране завршног/дипломског рада аутора: **02.07.2020.**

Наслов завршног/дипломског рада аутора:

Развој Андроид апликација коришћењем програмског језика Котлин

Академско звање коју је аутор стекао одбраном завршног/дипломског рада:

дипломирани инжењер електротехнике (240 ECTS)

Академско звање које је аутор стекао одбраном мастер/магистарског рада:
мастер електротехнике - 300 ECTS - Рачунарство и информатика

Назив факултета/Академије на коме је мастер/магистарски рад одбрањен:
Електротехнички факултет, Универзитет у Бањој Луци

Наслов мастер/магистарског рада и датум одбране:
Примјена машинског учења за описивање слика помоћу текста, 30.05.2025.

Научна област мастер/магистарског рада према CERIF шифрарнику: **T 120**

Имена ментора и чланова комисије за одбрану мастер/магистарског рада:

1. Проф. др Владимир Рисојевић, предсједник
2. Проф. др Зоран Ђурић, ментор
3. Доц. др Михајло Савић, члан

У Бањој Луци, дана 20.05.2025.

Декан

Изјава 1

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је

мастер/магистарски рад

Наслов рада: Примјена машинског учења за описивање слика помоћу текста

Наслов рада на енглеском језику: The Application of Machine Learning for Image Captioning

- ☑ резултат сопственог истраживачког рада,
- ☑ да мастер/магистарски рад, у цјелини или у дијеловима, није био предложен за добијање било које дипломе према студијским програмима других високошколских установа,
- ☑ да су резултати коректно наведени и
- ☑ да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

У Бањој Луци 22.05.2025.

Потпис кандидата

Александра Мићић

Изјава 2

Изјава којом се овлашћује Електротехнички факултет/ Академија умјетности Универзитета у Бањој Луци да мастер/магистарски рад учини јавно доступним

Овлашћујем Електротехнички факултет/ Академију умјетности Универзитета у Бањој Луци да мој мастер/магистарски рад, под насловом

Примјена машинског учења за описивање слика помоћу текста

који је моје ауторско дјело, учини јавно доступним.

Мастер/магистарски рад са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Мој мастер/магистарски рад, похрањен у дигитални репозиторијум Универзитета у Бањој Луци, могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице *(Creative Commons)*, за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство - некомерцијално - без прераде
4. Ауторство - некомерцијално - дијелити под истим условима
5. Ауторство - без прераде
6. Ауторство - дијелити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

У Бањој Луци 22.05.2025.

Потпис кандидата

Александра Митић

Изјава 3

Изјава о идентичности штампане и електронске верзије

мастер/магистарског рада

Име и презиме аутора Алексије Мутић

Наслов рада Примјена машинског учења за описивање слика помоћу текста

Ментор проф. др Зоран Бурић

Изјављујем да је штампана верзија мог мастер/магистарског рада идентична електронској верзији коју сам предао/ла за дигитални репозиторијум Универзитета у Бањој Луци.

У Бањој Луци 22.05.2025.

Потпис кандидата

Алексије Мутић

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Патре 5
78000 Бања Лука

Проф. др Владимир Рисојевић
Електротехнички факултет Универзитета у Бањој Луци

Проф. др Зоран Ђурић
Електротехнички факултет Универзитета у Бањој Луци

Доц. др Михајло Савић
Електротехнички факултет Универзитета у Бањој Луци

НАУЧНО-НАСТАВНОМ ВИЈЕЋУ

ЕЛЕКТРОТЕХНИЧКОГ ФАКУЛТЕТА УНИВЕРЗИТЕТА У БАЊОЈ ЛУЦИ

Одлуком Научно-наставног вијећа Електротехничког факултета Универзитета у Бањој Луци број 20/3.709-9/24 од 14.10.2024. године, именовани смо за чланове Комисије за завршни рад II циклуса, под називом "**ПРИМЈЕНА МАШИНСКОГ УЧЕЊА ЗА ОПИСИВАЊЕ СЛИКА ПОМОЋУ ТЕКСТА**", кандидата Алексија Мићића. Након прегледа приложеног рада подносимо сљедећи

ИЗВЈЕШТАЈ

1. БИОГРАФСКИ ПОДАЦИ КАНДИДАТА

Рођен је 06.03.1997. године у Новом Саду. Електротехнички факултет Универзитета у Бањој Луци уписао је 2015. године, студијски програм Рачунарство и информатика, смјер Софтверско инжењерство. Дипломирао је са просјеком оцјена 8,49. Дипломски рад на тему "Развој Андроид апликација коришћењем програмског језика Котлин" успјешно је одбранио 02.07.2020. године.

Тренутно је запослен у компанији „Bravo Systems" д.о.о. Бања Лука, гдје ради на позицији *Data Science* инжењера. Неки од важнијих пројеката на којима је учествовао су:

1. Развој рјешења машинског учења за препознавање странице са производом, те проналазак идентичних, сличних и повезаних производа за дати производ.
2. Унапређење постојећег рјешења за CPA (*Cost per Action*) оптимизацију, коришћењем техника обраде природног језика.

Студије II циклуса студија, студијски програм Рачунарство и информатика, уписао је школске 2023/2024 године, на којем је положио све испите са просјечном оцјеном 10.

Кандидат је до сада објавио један рад на научној конференцији међународног значаја:

- Mićić, Z. Đurić, "Primjena mašinskog učenja za opisivanje slika pomoću teksta," YU INFO 2025, Kopaonik, Srbija, 2025, rad prihvaćen za objavu u zborniku radova.

2. ОСНОВНИ ПОДАЦИ О РАДУ

Завршни рад II циклуса студија кандидата Алексија Мићића, под називом "**Примјена машинског учења за описивање слика помоћу текста**", садржи 83 нумерисане странице са укупно 35 слика и 11 табела, а организован је у осам глава:

1. Увод,
2. Машинско учење,
3. Неуронске мреже,
4. Трансформатори,
5. Обрада природног језика,
6. Претрага продавнице,
7. Експериментални дио и
8. Закључак

Списак коришћене литературе садржи 48 цитираних извора.

3. АНАЛИЗА РАДА

У уводном дијелу рада прво су описани мотивација за израду рада и предмет истраживања, а затим су наведени циљеви, методологија и остварени резултати истраживања. На крају је укратко приказан садржај рада по главама. Основну мотивацију за истраживање и израду овог завршног рада представља велики број изазова у рјешавању проблема проналаска сличних и повезаних производа код претраге е-продавнице.

У другом поглављу описане су теоријске основе из области машинског учења. Дате су дефиниције основних појмова, као и кратак осврт на поступак обучавања модела класификатора. У овом поглављу посебно су обрађене технике класификације и кластеризације.

У трећем поглављу су описане дубоке неуронске мреже, које се користе за рјешавање бројних проблема из области машинског учења, укључујући анализу сличности слика и детекцију објеката. Неуронске мреже чине основу архитектуре многих савремених модела за векторску репрезентацију текста. У овом поглављу описане су једноставне мреже без повратних веза, а дат је и опис конволуционих неуронских мрежа и рекурентних неуронских мрежа које се у спрези могу користити за генерисање описа слика.

У четвртом поглављу обрађена је архитектура трансформатора, која представља унапређење у односу на неуронске мреже описане у трећем поглављу. Објашњен је основни принцип њиховог рада, након чега је приказана архитектура и начин функционисања модела који се користе за описивање слика помоћу текста. На крају овог поглавља су описани велики језички модели, са посебним фокусом на мултимодалне велике језичке моделе, који омогућавају генерисање јако детаљних и квалитетних описа.

Пето поглавље се бави обрадом природног језика. Прво су описане различите врсте векторских репрезентација текста, почевши од најпростијих базираних на фреквенцији појављивања ријечи, до оних сложенијих базираних на уграђивању ријечи и контекстуалним језичким моделима. Након тога су описане стандардне метрике које се у литератури користе за евалуацију квалитета генерисаних описа слика, при чему је дат осврт на њихове врлине и мане.

У шестом поглављу описан је проблем претраге продавнице производа. Прво је описана и објашњена архитектура традиционалног система за претрагу продавнице

производа, која се базира на детекцији објеката и анализи сличности слика. Наведени су и илустровани најважнији недостаци традиционалног приступа, гдје се разликују недостаци који потичу од модела за детекцију објеката и недостаци који потичу од модела за анализу сличности слика. Потом је дат опис предложеног рјешења које комбинује традиционални приступ базиран на анализи сличности слика са описима слика генерисаним помоћу техника машинског учења.

У седмом поглављу су описани експериментални резултати рада. Објашњен је поступак обучавања модела за традиционални приступ претраге производа, дат је и опис кориштених модела и скупова података за овај дио система. Потом су описани скупови података, као и модели који су кориштени за проблем генерисања описа слика одјевних предмета. Такође, описан је и процес *prompt engineering*-а који се користи код великих језичких модела у циљу добијања што квалитетнијих одговора. Затим су дати резултати евалуације квалитета описа на метрикама које су претходно описане у петом поглављу. На крају овог поглавља дати су резултати евалуације примјене комбинованог приступа на проблемима претраге који су претходно описани у шестом поглављу рада.

На крају рада дата су закључна разматрања и приједлози могућих унапређења, као и правци будућег истраживања.

4. НАЈВАЖНИЈИ ДОПРИНОСИ

Комисија сматра да је кандидат, кроз спроведено истраживање, реализовао завршни рад II циклуса студија који садржи више значајних доприноса, од којих су најважнији сљедећи:

1. Извршена је детаљна анализа проблема који се јављају у случајевима примјене традиционалног приступ за претрагу производа у е-продавници,
2. Извршена је детаљна анализа свих теоријских концепата из области машинског учења, а који су неопходни за рјешавање описаних проблема,
3. Дат је приједлог једног приступа за рјешавање описаних проблема, а који комбинује примјену модела за анализу сличности слика заједно са моделом за описивање слика, тј. гдје се врши линеарно комбиновање векторске репрезентације добијене на основу визуелних карактеристика слике са векторском репрезентацијом генерисаног описа слике, и
4. Извршен је већи број практичних експеримената којима је показано да предожени приступ успјешно рјешава описане проблеме који се јављају у случајевима примјене традиционалног приступа за претрагу производа у е-продавници.

5. ЗАКЉУЧАК И ПРИЈЕДЛОГ

Комисија сматра да завршни рад II циклуса под називом "**ПРИМЈЕНА МАШИНСКОГ УЧЕЊА ЗА ОПИСИВАЊЕ СЛИКА ПОМОЋУ ТЕКСТА**", кандидата Алексија Мићића, садржи све потребне елементе и резултате којима су остварени постављени циљеви истраживања, те са задовољством предлаже Научно-наставном вијећу Електротехничког факултета Универзитета у Бањој Луци да усвоји извјештај Комисије и одобри заказивање усмене јавне одбране.

Бања Лука, 05.05.2025. године

1. Проф. др Владимир Рисојевић, предсједник

2. Проф. др Зоран Бурић, ментор

3. Доц. др Михајло Савић, члан