SVEUČILIŠTE U RIJECI
FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Tedo Vrbanec

OTKRIVANJE PLAGIRANJA PRI PARAFRAZIRANJU KORIŠTENJEM MODELA DUBOKOG UČENJA

DOKTORSKI RAD

Rijeka, 2025.

SVEUČILIŠTE U RIJECI
FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Tedo Vrbanec

OTKRIVANJE PLAGIRANJA PRI PARAFRAZIRANJU KORIŠTENJEM MODELA DUBOKOG UČENJA

DOKTORSKI RAD

Mentorica: prof. dr. sc. Ana Meštrović

Rijeka, 2025.

UNIVERSITY OF RIJEKA
FACULTY OF INFORMATICS AND DIGITAL TECHNOLOGIES

Tedo Vrbanec

THE DETECTION OF PARAPHRASING PLAGIARISM BY USING DEEP LEARNING MODELS

DOCTORAL THESIS

Mentor: prof. dr. sc. Ana Meštrović

Rijeka, 2025.

Mentorica: prof. dr. sc. Ana Meštrović

Doktorski rad obranjen je dana _____ na Fakultetu informatike i digitalnih tehnologija, pred povjerenstvom u sastavu:

1. _____
2. _____
3. _____
4. _____
5. _____

Sažetak

S obzirom na mnogovrsnost tipova plagijata te načina i složenosti njihova stvaranja, otkrivanje akademskog plagiranja kompleksan je zadatak. Za (raz)otkrivanje jednostavnijih vrsta plagijata i utvrđivanje originalnosti tekstova danas postoje pouzdani i efikasni postupci, metode, algoritmi i pretežito komercijalni programski sustavi, no neke metode plagiranja i njihove kombinacije rezultiraju sofisticiranim i prikrivenim plagijatima čije je otkrivanje vrlo teško i za koje još nema algoritamskog rješenja problema njihova pronalaska. Ipak, razvijanjem odgovarajućih pristupa, metoda i algoritama nastoje se postići programska rješenja koja će otkrivanje i takvih, prikrivenih plagijata učiniti rutinski rješivim, računalno podržanim zadacima. Ovo istraživanje predstavlja pristup koji tekstove promatra i uspoređuje ih na semantičkoj razini. Sredstvo kojim se u ovome istraživanju koristimo za pristup semantici tekstova jesu predtrenirani veliki jezični modeli koji nastaju kao rezultat treniranja dubokih neuronskih mreža velikom količinom tekstova. Tako dobiven opis semantike u obliku vektorskih reprezentacija jezičnih modela ipak nije dovoljan. Potrebna je i mjera sličnosti ili udaljenosti koja se računa između vektorskih veličina, pa je u ovome istraživanju provedena empirijska analiza s ciljem odabira mjere koja daje najbolje rezultate. Pored modela i mjere sličnosti istraživanje je identificiralo i mogućnost uspješne kombinacije s mjerama sličnosti i veličine tekstova koji se uspoređuju na razini riječi. Konačno, potrebni su i označeni korpusi parafraziranih tekstova. Cilj istraživanja bio je razvijanje i implementiranje novog postupka za utvrđivanje plagiranja pri parafraziranju kao važnoj vrsti prikrivenog plagiranja, pri čemu su ostvareni znanstveni doprinosi u: (i) oblikovanju korpusa parafraziranih dokumenata pogodnog za učenje i evaluaciju postupaka otkrivanja plagiranja pri parafraziranju, (ii) razvoju i implementaciji novog postupka za otkrivanje plagiranja pri parafraziranju temeljenog na modelu dubokog učenja, (iii) definiranju postupka i mjere evaluacije parafraziranja tekstova, (iv) utvrđivanju tehnika obrade teksta koje pozitivno utječu na otkrivanje parafraziranja, (v) utvrđivanju primjerenosti različitih mjera sličnosti za računanje sličnosti vektorskih reprezentacija tekstova, (vi) utvrđivanju složenosti metoda korištenih u zadatku utvrđivanja da se radi o parafraziranom tekstu, (vii) utvrđivanju optimalnih graničnih vrijednosti za detekciju sličnosti i parafraziranja, (viii) usporedbi performansi metoda u zadatku detekcije parafraziranja temeljenih na korpusu i (ix) definiranju formule kojom se aproksimira optimalni broj dimenzija vektorskih reprezentacija jezičnih modela dubokog

učenja u zadatku detekcije parafraziranja.

Ključne riječi: semantička sličnost, duboko učenje, parafrazirani korpus, prikriveni plagijati, parafraziranje, mjera parafraziranosti, jezični modeli, vektorske reprezentacije.

Abstract

Considering the diversity of plagiarism types and the complexity of their creation, detecting academic plagiarism is a complex task. Reliable and efficient procedures, methods, algorithms, and predominantly commercial software for determining the originality of texts already exist for detecting simpler types of plagiarism. However, some methods of plagiarism and their combinations result in sophisticated and concealed forms of plagiarism, whose detection is very difficult and for which there is still no solution, i.e., no algorithmic solution for their detection currently exists, making their identification through software solutions impossible. Nevertheless, approaches, methods, and algorithms that promise progress in detecting these types of plagiarism are being developed, which suggests a convergence of the problem of plagiarism detection towards a category of routine, computer-supported solutions. This research presents an approach that examines and compares texts on a semantic level. The tools used in the research to extract the semantic information from texts are pre-trained large language models, which are created as a result of training deep neural networks on vast amounts of text data. However, the semantics derived in the form of vector embeddings of language models are not sufficient on their own. A standard similarity or distance measure is required to compute the relationships between vector representations. In this research, an empirical analysis was conducted to identify the measure that yields the best results. In addition to the model and similarity measure, the research identified their successful combination with similarity and text size measures, which operate on the word level. Finally, annotated corpora of paraphrased texts are also required. The aim of this research was to develop and implement a new procedure for detecting plagiarism in paraphrasing, which is a significant form of concealed plagiarism. The study's scientific contributions include: (i) designing a corpus of paraphrased documents suitable for training and evaluating plagiarism detection procedures, (ii) developing and implementing a novel procedure for detecting plagiarism through paraphrasing based on a deep learning model, (iii) defining procedures and evaluation measure for text paraphrasing, (iv) identifying text processing techniques that positively impact paraphrase detection, (v) determining the suitability of various similarity measures for calculating the similarity of vector representations of texts, (vi) assessing the complexity of methods used in the task of determining text paraphrasing, (vii) establishing optimal threshold values for binarizing similarity scores obtained by different methods in the

task of paraphrase detection, (viii) assessing the performance of corpus-based methods in the task of paraphrase detection, and, (ix) defining a formula to approximate the optimal number of dimensions for vector representations of deep learning language models in the task of paraphrase detection.

Keywords: semantic similarity, deep learning, paraphrase corpora, concealed plagiarism, paraphrasing, paraphrase measure, language models, vector embeddings.

Sadržaj

Kazalo slika

Kazalo tablica

1. Uvod

Plagiranje je pripisivanje tuđih zasluga za neki intelektualni, umjetnički ili materijalni rad, odnosno proizvod (Ali i Taqa, 2023; Al-Shamery i ALkhafaji, 2017). Smatra se nepoštenim, neetičnim i nemoralnim postupkom. Stoga je tako utvrđeno djelovanje pod udarom propisa i zakona. Plagiranje se, vjerojatno, javlja otkako je ljudskog društva, a moguće je smatrati da je ono u najvećem dijelu povijesti bilo i poticajno za njegov razvoj. Ipak, moderno društvo normativno je reguliralo pripisivanje zasluga autorstva konkretnim osobama, radnim ili projektnim skupinama, financijerima (tvrtkama, agencijama, institutima, državnim institucijama) putem prijave i registracije patentnih prava, prava korištenja imena te putem propisa kojima se štiti pravo intelektualnog vlasništva, umjetničkog i dizajnerskog izričaja kao i prvotnost objave ideja i znanstvenih otkrića. Pojavom interneta informacije postaju lako dostupne i plagiranje je olakšano – međutim olakšano je i uočavanje plagijata, pa se stječe dojam eksponencijalno rastućeg problema, posebice ondje gdje je to najviše nedolično jer utječe na osobnu i profesionalnu evaluaciju rada i postignuća pojedinaca i društva: u znanosti, visokom obrazovanju, izdavaštvu i medicini.

U akademskoj domeni glavni način diseminacije znanja, otkrića i ideja odvija se putem digitalnih tekstnih dokumenata (Abdelhamid i sur., 2022), akademskih radova (članaka u časopisima, teza u obliku monografija, knjiga, izvještaja, izvješća s izlaganja na znanstvenim i stručnim skupovima), pa se njihovo plagiranje stoga naziva akademsko. Plagiranje korištenjem manjih ili većih istovjetnih dijelova teksta lako se otkriva specijaliziranim programskim alatima (Vrbanec i Meštrović, 2017), no kako otkrivanje prikrivenoga plagiranja ostaje još uvijek neriješen problem, ovo je istraživanje posvećeno upravo tome izazovu – unaprjeđivanju automatskih postupaka otkrivanja plagijata u akademskome tekstu.

Dva su pristupa pronalasku sličnosti ili istovjetnosti kod otkrivanja plagijata u akademskom tekstu: jedan se odnosi na otkrivanje plagiranja *izražavanja* ideja (lat. *forma*), a drugi na otkrivanje plagiranja samih *ideja* (grč. *idea*). Prvi je pristup tehnički i logički lakše izvediv, pa je u njemu dosad ostvaren veći razvoj kako u teorijskom smislu (algoritmi, metode) (Wang i Dong, 2020), tako i u programskim sustavima za njihovo otkrivanje. No taj pristup nije i neće biti u mogućnosti otkriti sve plagijate, osobito one prikrivene: koji nastaju

prijevodom, korištenjem istoznačnica, parafraziranjem, sažimanjem, kompilacijom ili kombinacijom više metoda plagiranja. Uvidjevši ograničenja toga pristupa suvremeni se istraživači sve više okreću drugome koji je mnogo složeniji i od kojeg se tek očekuje ostvarenje punog potencijala (Amur i sur., 2023; Babić i sur., 2020, 2020; Han i sur., 2021; Mahmoud i Zrigui, 2019). Kako se istraživači okreću metodama prepoznavanja semantike teksta, usporedba tekstova pomiče se s površinske razine na višu, semantičku razinu, tj. u traganje za semantičkom sličnošću, stilskom nedosljednošću, uspoređivanjem značenja i sl. Riječ je o algoritamski i programski složenijim zadacima, a i zahtjevnijima u pogledu potrebne računalne snage, za koju je sve češće potrebno pokretati distribuirane mrežne operacijske i programske sustave, resurse računalnih klastera i oblaka kako bi se provjera plagiranja provela u razumnom vremenu. Uspješnošću otkrivanja semantičke sličnosti između dokumenata otvorila bi se mogućnost otkrivanja značajno većeg postotka složenijih, profinjenijih i prikrivenih vrsta plagijata, za koje danas nema zadovoljavajućih rješenja izvan domene ljudskog eksperta koji će, bez obzira na napredak automatiziranih sustava za pronalaženja plagijata i dalje ostati konačni prosuditelj je li određeni dokument doista plagijat ili nije.

1.1. Ciljevi istraživanja

Opći cilj istraživanja je razvoj metode za otkrivanje prikrivenih plagijata nastalih kao rezultat manipulacija tekstom tehnikom parafraziranja, vrstom prikrivenog plagiranja u čiju kategoriju još pripadaju plagiranje prijevodom, korištenje istoznačnica, sažimanje, kompilacija ili kombinacija više metoda plagiranja. Parafraziranje je proces prepisivanja teksta kako bi se promijenio njegov oblik i izraz, zadržavajući izvorno značenje (Vrbanec i Meštrović, 2020). Automatsko otkrivanje parafraziranja ima važnu ulogu u raznim zadacima, poput otkrivanja plagijata, atribucije autorstva, odgovaranja na pitanja, sažimanje teksta, rudarenje teksta itd. (Vrbanec i Meštrović, 2023; Wahle i sur., 2023; C. Zhou i sur., 2022).

Pri otkrivanju plagijata u ovome se istraživanju primjenjuje postupak ekstrinzičnog otkrivanja plagiranja korištenjem vanjskoga korpusa dokumenata s kojim se vrši usporedba. Ekstrinzično otkrivanje plagiranja podrazumijeva uspoređivanje teksta s vanjskim izvorima kako bi se identificirali dijelovi koji su potencijalno plagirani. Vanjski izvori mogu biti znanstvene baze članaka, opći ili specijalizirani pretraživači, mrežne stranice, knjige i

akademski radovi u digitalnom obliku iz nekoga repozitorija u slobodnom dostupu i sl. Specijalizirana programska podrška poput Turnitin, iThenticate ili Grammarly koriste napredne algoritme za usporedbu tekstova s brojnim vanjskim izvorima. S druge strane, intrinzično otkrivanje plagiranja podrazumijeva analizu teksta bez oslanjanja na vanjske izvore, s ciljem identificiranja promjena u stilu pisanja, vokabularu, strukturi, stilu citiranja i sl., a koje mogu ukazivati na plagiranje.

Za ostvarenje općeg cilja potrebno je ostvariti mnoštvo specifičnih ciljeva. Primjerice, za otkrivanje plagijata nastalih parafraziranjem nije dostatno pronalaženje n-grama iste klase (potpuno istih n-grama, korištenje istoznačnica ili restrukturiranje teksta) i zato je specifični cilj pronalazak načina *prepoznavanja semantike* teksta sadržanog u nekom dokumentu te pronalazak načina *usporedbe semantike* između dokumenata. U tom je smislu obećavajući pristup onaj koji koristi jezične modele dubokog učenja za reprezentaciju teksta i mjerenje stupnja semantičke sličnosti dokumenata te parafraziranja na razini rečenica. . Za otkrivanje plagiranja parafraziranjem potrebna je računalna programska podrška, a njezin je razvoj također jedan od specifičnih ciljeva ovog istraživanja bez kojega istraživanje ne bi bilo moguće ostvariti.

1.2. Motivacija

U akademskoj se zajednici za *online* podršku nastavi koriste razni sustavi za upravljanje učenjem (engl. *Learning Management System*, LMS), a jedan od poznatijih među njima je Moodle. U sklopu nastavnih obaveza predviđenih nastavnim planom i programom studenti često izrađuju zadaće ili seminare u kojima istražuju zadane ili odabrane teme. Svoje radove predaju putem LMS-ova na pregled i ocjenu nastavnicima. Slično vrijedi i za završne te diplomske radove. Nadalje, akademska zajednica organizira konferencije, stručne i znanstvene skupove, publicira znanstvene i stručne časopise. Za njih najčešće koristi sustave za podršku publiciranju, a među njima je primjerice *Open Journal Systems* (OJS). Obje vrste sustava u stanju su koristiti razne dodatke, poput onih za provjeru plagiranja/originalnosti. U prošlosti su ti dodaci postojali i bili su otvorenoga koda. Vremenom su napuštani ili komercijalizirani, pa su do danas iščeznuli. No, kada bi ih i bilo, plagijatori sve češće koriste prikriveno plagiranje, tako da su čak i postojeći komercijalni sustavi, iako prilično skupi za akademsku zajednicu, postali nedovoljno učinkoviti u otkrivanju prikrivenih plagijata.

U posljednjem desetljeću područje obrade prirodnog jezika značajno je napredovalo razvojem dubokih neuronskih mreža i arhitektura poput transformera. Stoga je važno istraživati postupke koji se temelje na modelima dubokog učenja za otkrivanje plagijata koji nastaju parafraziranjem.

U kontekstu akademske zajednice i otkrivanja plagiranja kao teške povrede akademske čestitosti, pronalazak postupaka i metoda utvrđivanja semantičke sličnosti tekstova jedan je od prioritetnih zadataka, a koji je upravo u fokusu ovoga istraživanja. Budući da je pojam plagijata širok, ovo je istraživanje posvećeno problemu parafraziranja kao obliku prikrivenog plagiranja. Ipak, gotovo je sigurno da će se dobiveni rezultati moći ekstrapolirati te koristiti i za otkrivanje drugih vrsta prikrivenih plagijata.

1.3. Hipoteze

U istraživanju su postavljene dvije istraživačke hipoteze:

H1: Primjenom modela dubokog učenja moguće je otkrivati plagiranje pri parafraziranju teksta.

H2: Odgovarajućom kombinacijom modela dubokog učenja i različitih tehnika pripreme teksta moguće je poboljšati otkrivanje parafraziranja.

1.4. Očekivani znanstveni doprinosi

Očekivani su i ostvareni sljedeći znanstveni doprinosi istraživanja:

- oblikovan je korpus dokumenata pogodan za učenje i evaluaciju postupaka za otkrivanje plagiranja pri parafraziranju
- razvijen je i implementiran novi postupak za otkrivanje plagiranja pri parafraziranju, zasnovan na modelu dubokog učenja
- definiran je postupak i mjera evaluacije parafraziranja tekstova.

Pored navedenih i očekivanih, ostvareni su i dodatni znanstveni doprinosi:

- utvrđene su performanse metoda u zadatku detekcije parafraziranja temeljenih na korpusu
- utvrđene su optimalne granične vrijednosti za određivanje sličnosti dokumenata i parafraziranja na razini rečenica

- utvrđena je složenost korištenih metoda u O notaciji
- utvrđena je primjerenost (rang-lista uspješnosti prema F-mjeri) različitih mjera sličnosti i udaljenosti za računanje sličnosti vektorskih reprezentacija (engl. *vector embeddings*) tekstova
- određena je funkcija kojom se aproksimira optimalna dimenzija vektorske reprezentacije teksta dobivene učenjem jezičnih modela temeljenih na dubokom učenju
- utvrđene su tehnike obrade teksta koje pozitivno utječu na otkrivanje parafraziranja.

1.5. Struktura rada

Nakon uvodnog poglavlja, u kojem su navedeni ciljevi istraživanja, motivacija za istraživanje, hipoteze te znanstveni doprinosi istraživanja, u drugom se poglavlju predstavlja objekt istraživanja – plagijat: definicije plagijata i plagiranja, uzroci i posljedice modernoga plagiranja, posebno akademskoga plagiranja, taksonomija i metode nastanka akademskih plagijata te su prikazani pristupi otkrivanju plagiranja. U nastavku toga poglavlja iznesen je pregled dosadašnjih istraživanja i njihovi nedostaci te je opisan dosadašnji razvoj programske podrške za otkrivanje plagiranja. U trećem poglavlju predstavljene su metode iz područja obrade prirodnog jezika koje su relevantne za ovo istraživanje. Predstavljeni su postojeći statistički pristupi, probabilistički pristup, pristup vektorskim reprezentacijama teksta kao rezultata treniranja jezičnih modela temeljenih na dubokom učenju, veliki jezični modeli te inherentna uključenost semantike teksta u modelima dubokog učenja. Nadalje, prezentirane su metode mjerenja sličnosti, odnosno udaljenosti tekstova te mjere evaluacije. Četvrto se poglavlje bavi metodološkim temeljem istraživanja koje uključuje razvoj i formalni prikaz nove metode temeljene na dubokom učenju za otkrivanje parafraziranja, korištenje postojećih i stvaranje vlastitih označenih korpusa parafraziranih tekstova nužnih za istraživanje. Također je predstavljena i nova kompozitna mjera parafraziranosti pomoću jezičnih modela dubokog učenja. U petom poglavlju prikazana je implementacija istraživanja i provedenih eksperimenata, s tehničkim i sistemskim specifikacijama, podatkovnim skupovima i njihovim podjelama, eksperimentima nad cjelovitim dokumentima i nad rečenicama kao osnovnim logičkim cjelinama, a poglavlje završava analizom složenosti pojedinih metoda. Šesto poglavlje prikazuje rezultate i njihovu evaluaciju. U sedmom su poglavlju raspravljeni

rezultati, utvrđeni znanstveni doprinosi te je objašnjeno potvrđivanje hipoteza. Nakon zaključnoga, osmog poglavlja i popisa korištene literature slijedi sedam dodataka koji nisu dio integralnoga teksta rada, a koji su važni za potvrđivanje sažetih rezultata ili koji objašnjavaju neke tvrdnje iznesene u glavnom dijelu rada.

2. Plagijati

Prvi radovi o plagiranju tekstova i izvornog programskog koda datiraju iz 1970-ih godina (Alzahrani i sur., 2012). U njima se pretežito izvješćuje o otkrivanju plagijata uz pomoć izvornih programa pisanih u programskim jezicima *Pascal* i *C*. Dvadesetak godina poslije pojavili su se radovi u kojima su predstavljene statističke računalne metode otkrivanja kopiranja u prirodnim jezicima. 1990-ih godina znanstvenici su počeli u većoj mjeri objavljivati radove o akademskim plagijatima, pa je tako Samuelson polemizirao o etičnosti i kršenju autorskih prava izdavača u slučaju autoplagiranja (Samuelson, 1994). Autori na prijelazu tisućljeća uglavnom su se bavili problemima pronalaženja plagijata u zatvorenim sustavima unutar akademskih ustanova i mrežnim plagiranjem. Suvremeni istraživači pokušavaju (1) dotjerati postojeće sustave kako bi oni bili učinkoviti, (2) koriste semantičke i stilističke sličnosti dokumenata i (3) pronalaze načine strojnog „razumijevanja“ značenja teksta.

Vjerojatnost da dvije osobe bez međusobnog utjecaja napišu identičan netrivijalan tekst ili naprave identično netrivijalno djelo vrlo je mala, no neka istraživanja, poput (Alzahrani i sur., 2012), tvrde da je to i nemoguće. Postupak preuzimanja tuđih misli, riječi ili djela bez jasne naznake izvora naziva se plagiranje, a proizvod plagijat. Plagijat (lat. *plagiare* = ukrasti; lat. *plagere* = oteti; lat. *plagiarius* = otmičar) je djelomično ili u cijelosti preuzet tuđi intelektualni ili umjetnički rad, bez jasne naznake tuđega autorstva. Korištenje tuđega djela najčešće nije nedozvoljeno i neetično, ali postoje norme na koji način takvo korištenje treba i označiti (Clough, 2003), a ponekad i zatražiti pismeno dopuštenje autora. Problem plagiranja u akademskoj zajednici i istraživačkim organizacijama dodatno je učinila važnim i zanimljivim laka dostupnost dokumenata i informacija u internetsko doba (Kumar i Tripathi, 2013).

Plagiranje je u većini država zakonom zabranjeno i sankcionirano (Green, 2002; Kumar i Tripathi, 2013). Prema hrvatskom Zakonu o autorskom pravu i srodnim pravima

(Hrvatski Sabor, 2021): „autorsko pravo pripada, po svojoj naravi, fizičkoj osobi koja stvori autorsko djelo". Svjetska organizacija za intelektualno vlasništvo (WIPO) donijela je 1886. godine Bernsku konvenciju (Turnitin Europe, 2016; World Intellectual Property Organization (WIPO), 1886), koju je do 2024. god. ratificirala 181 zemlja, a koja postavlja standarde zaštite, ali i prava korištenja autorskih djela u državama potpisnicama. Odnos prema plagiranju i plagijatorima sve je stroži, pa se plagiranje sve više približava kategoriji računalnoga kriminala, zajedno s računalnim virusima, hakiranjem, slanjem neželjene pošte i kršenjem autorskih prava (Al-Shamery i Gheni, 2016).

2.1. Definicije

Postoji visoko suglasje o definiciji pojma plagijata. Većina izvora, poput (Culwin i Lancaster, 2001a; Kumar i Tripathi, 2013; Lancaster, 2003; Lukashenko i sur., 2007; Zu Eissen i Stein, 2006), koristi se, uz manje varijacije, definicijom koja se nalazi i u rječniku (Merriam-Webster Dictionary, 2016), a koja definira plagijat kao djelo nastalo korištenjem tuđih riječi ili ideja bez priznavanja zasluga izvornom autoru. *Encyclopaedia Britannica* definira plagijat kao čin uzimanja spisa druge osobe te njegovu predaju kao vlastitog. (Encyclopaedia Britannica, 2024). Rječnici izdavačkih kuća *Cambridge University Press* i *Oxford* definiraju plagijat kao korištenje ideja ili rada drugih osoba pretvarajući se da su vlastita (Cambridge University Press, 2018; Oxford Dictionary, 2018). Meuschke i Gipp (2013) definiraju akademski plagijat kao preuzimanje tuđih ideja ili izričaja bez davanja dužnog priznanja izvornim autorima ili izvorima prema akademskim načelima (Meuschke i Gipp, 2013). Na stranici *Plagiarism.org* smatra se da su plagiranje i plagijat vrsta prijevare: uključuju krađu tuđega djela te potom laganje o toj krađi (Plagiarism.org, 2017). Prema međunarodnoj organizaciji *Aktion Plagiarius*, plagijat je imitacija proizvoda u svrhu gospodarskoga korištenja (Aktion Plagiarius, 2018). Jeremy B. Williams (2005) smatra da je plagiranje „oblik varanja koji se općenito smatra moralno i etički neprihvatljivim" (J. B. Williams, 2005).

Uz plagijat se veže nekoliko srodnih pojmova (Aktion Plagiarius, 2018): krivotvorina, piratiziranje dizajna, piratiziranje robne marke (engl. *brand*), replika i kršenje autorskih prava.

- **Krivotvorina** ili imitacija je proizvod koji krivotvoritelj potencijalnom kupcu predstavlja kao original, dakle kupca se nastoji uvjeriti da je riječ o originalnom

proizvodu. Krivotvorenje je kazneno djelo.

- **Piratiziranje dizajna** je marketinški koncept kojim se proizvođači koriste kako bi u kratkom vremenu iskoristili veliko zanimanje kupaca za određeni proizvod na način da dizajn njihova proizvoda jako podsjeća na neku poznatu marku.
- **Piratiziranje robne marke** je situacija kada proizvođač ne može zaštititi svoje ime i proizvode u nekoj zemlji jer je to prethodno učinio netko drugi s kime je nužno postići financijski sporazum.
- **Replika** je nova izrada određenoga proizvoda, koju može izvršiti izvorni proizvođač ili vlasnik prava.
- **Kršenje autorskih prava** je „intenzivno korištenje nečijeg rada bez dozvole, sa ili bez priznanja" tuđeg autorstva (Chaudhuri, 2008).

Plagijati se često koriste u poslovnom svijetu kako bi se: (a) bez većega ulaganja došlo do novijih proizvoda, dok još imaju znatnu profitabilnost, (b) na nezakonit način iskoristila tuđa robna marka (engl. *brand*) ili (c) okoristilo tuđim dizajnom ili idejom. Ekonomske posljedice industrijskih plagijata su teške, a procjene (Aktion Plagiarius, 2018) govore da 10% svjetske trgovine čine krivotvorine i plagijati te da se godišnje zbog toga izgubi 200-300 milijardi eura i 200 tisuća radnih mjesta.

Akademski plagijati, odnosno digitalni tekstni plagijati najčešći su objekt plagiranja tijekom obrazovanja te u akademskim radovima. Akademsko plagiranje sintagma je koja označava plagiranje – u cjelini ili u dijelovima – digitalnih tekstnih dokumenata sljedećih vrsta: programa u izvornome programskom kodu, seminara, kritičkih osvrta, stručnih i znanstvenih radova te neknjiževnoumjetničkih tekstova. Prvi tagmem sintagme – akademsko, naznačuje da se ta vrsta plagiranja najčešće pojavljuje u akademskoj zajednici. Naziv „akademsko" upozorava na to da je riječ o domeni u kojoj se čovjek lako i često suočava s pitanjem je li nešto plagiranje ili nije, odnosno rad u toj domeni zahtijeva od čovjeka veliku budnost kako bi se izbjegla opasnost od skretanja u plagiranje. Najsažetije kazano, „plagiranje je zločin prema akademskoj zajednici" (Bouville, 2008).

Prikriveno plagiranje čije je otkrivanje (u obliku parafraziranja) u srži ovoga istraživanja, takav je oblik plagiranja koji uključuje postupke kojima plagijator nastoji ostaviti dojam originalnosti, pri čemu može koristiti raznolike manje ili više inovativne postupke poput: promjene redoslijeda riječi, korištenje drukčijih izraza ili sinonima, promjene strukture rečenice, prevođenja teksta između dvaju ili više jezika i sl., pri čemu plagijator ne dodaje

vlastiti originalni doprinos u značajnoj mjeri te se ne referira na izvorne radove i autore.

2.2. Uzroci i posljedice modernog plagiranja

Prije europskoga prosvjetiteljstva (koje se odvijalo u 17. i 18. stoljeću), koncept autorskoga prava i intelektualnoga vlasništva nije bio razvijen u svijesti ljudi. "Posuđivanje", prepisivanje ili ponovno korištenje tuđih djela nije se smatralo moralno upitnim. Štoviše, širenje ideja bilo je važnije od jasno utvrđenog autorstva, pa se citiranje ili čak preuzimanje bez navođenja autora često doživljavalo kao legitimno i korisno (Joy i sur., 2009). Prosvjetiteljstvo je pak označilo prekretnicu u shvaćanju plagiranja. Povijesni, pravni i filozofski kontekst iz kojega izranja promjena u shvaćanju plagiranja sastoji se u sljedećem: prosvjetiteljstvo stavlja naglasak na originalnost, kreativnost i autonomiju pojedinca, izdavaštvo postaje profitabilno i stoga autori počinju prepoznavati financijski interes u zaštiti vlastitog djela, moralna filozofija prosvjetiteljstva uključuje pitanje odgovornosti i poštenja prema radu. Zahvaljujući novome gledištu plagiranje prelazi, iz prakse koja je nekoć smatrana normalnom, u društveno nepoželjnu i moralno neprihvatljivu pojavu. Takav je odnos prema autorstvu prisutan do danas. Pomak iz analognog u digitalno društvo, pojava interneta, a potom i *weba* 1990-ih, rezultirala je da su dokumenti i informacije postali lako dostupni. Prema rezultatima istraživanja (Turnitin Europe, 2016), plagiranje u obrazovnom sustavu, akademskoj zajednici i u istraživačkim organizacijama u stalnom je porastu (u smislu da apsolutna većina studenata i učenika primjenjuje plagiranje). Trend je potvrđen novijim izvještajima Turnitina iz 2020. i 2023. godine, koji potvrđuju da plagiranje u obrazovnom sustavu, akademskoj zajednici i istraživačkim organizacijama i dalje raste, osobito zbog porasta *online*-učenja i korištenja generativne umjetne inteligencije koja olakšava akademsko nepoštenje (C. Lee, 2020; Turnitin, 2023). Također, analiza objavljena u časopisu *Education and Information Technologies* 2021. pokazuje da su tijekom pandemije COVID-19 studenti povećali učestalost plagiranja zbog olakšanog pristupa vanjskim izvorima i tehnologijama tijekom *online*-ispita (Eshet, 2024). Dodatno se, prema izvješću u uglednom znanstvenom časopisu *Nature* iz 2023. godine, pojavila uporaba alata umjetne inteligencije poput ChatGPT-a za stvaranje lažnih, ali uvjerljivih znanstvenih radova, što dodatno komplicira borbu protiv plagiranja u akademskim krugovima (Taloni i sur., 2023).

Iako je oduvijek postojalo, plagiranje prije internetskog doba nije bilo toliko vidljivo,

pa ni sankcionirano. U današnje doba problematika plagiranja postala je vrlo važna i zanimljiva akademskoj zajednici i znanstvenim istraživačima (Kumar i Tripathi, 2013; Park, 2004; Roig, 2006), posebno zato što je više od 10 % radova u akademskoj zajednici nedovoljno originalno, tj. sadrži više od 50 % neizvornoga sadržaja (Turnitin Europe, 2016). Akademski plagijati mogu sadržavati vrlo različitu količinu plagiranog sadržaja (Joy i sur., 2009), od nekoliko rečenica pa do cijelih dokumenata podmetnutih kao vlastitih. Svoju su poslovnu nišu pronašli i svoje poslovanje razvili proizvođači akademskih plagijata po narudžbi, a Chong (2013) navodi desetak takvih svjetski poznatih slučajeva otkrivenih plagijata (M. Y. M. Chong, 2013).

Već su početkom tisućljeća neka istraživanja ukazivala (J. B. Williams, 2005) da oko 90 % studenata vara tijekom studija, a isti autor razlikuje tri tipa plagijatora: lijene, lukave i nenamjerne te zastupa ideju da su oni potonji često žrtve nedostatka akademskog iskustva, kulture i etičnosti. Učenici i studenti tijekom svojega se obrazovanja koriste plagiranjem za izradu programskih rješenja, zadaća ili seminara, a razlozi za to mogu biti: nedostatak vremena, sposobnosti ili motivacije, lijenost ili neznanje da je takvo postupanje nemoralno, nedopušteno i kažnjivo.

Otkriveni plagijati u pravilu nose značajne i dalekosežne negativne posljedice za plagijatore (Clough, 2000; Turnitin Europe, 2016) u obliku: javne sramote, financijskih kazni, izbacivanja iz obrazovnih institucija, poništenja diploma, gubitka radnoga mjesta, sudskih postupaka i osuda, negativnih ili smanjenih ocjena te otežanoga polaganja ispita.

Uzroci plagiranja u poslovnom svijetu uglavnom su vezani uz namjeru da se bez većeg ulaganja dođe do novih proizvoda dok oni još imaju znatnu profitabilnost, da se na nezakonit način iskoristi tuđa robna marka ili da se pribavi korist od tuđega dizajna ili ideje.

2.3. Taksonomija akademskih plagijata

Autori različitih istraživanja razlučuju više tipova akademskih plagijata, pri čemu postoje velike razlike u dubini i širini njihova pristupa. Jednu od prvih podjela postupaka akademskog plagiranja načinio je Martin (1994), koji razlikuje doslovno kopiranje, parafraziranje, plagiranje iz sekundarnih izvora, plagiranje strukture rada, plagiranje ideja i plagiranje autorstva (Martin, 1994).

Parc (2004) navodi pet tipova plagijata: suradnja ili dogovor (engl. *collusion*) – jedan

autor prisvaja zasluge grupe, narudžba (engl. *commission*) – dogovorna predaja tuđeg rada, dupliciranje ili ponavljanje (engl. *duplication*) – isti rad u dva različita konteksta, kopiranje/parafraziranje (engl. *copying*/*paraphrasing*) – preoblikovanje tuđeg rada i predaja tuđeg rada (engl. *submission*) – predaja tuđeg rada bez znanja izvornog autora (Park, 2004).

Maurer i suradnici (2006) plagijate dijele u četiri kategorije, ovisno o namjeri plagijatora: slučajni, nenamjerni, namjerni i autoplagijat (Maurer i sur., 2006).

Schwarzenegger i Wohlers (2006) razlikuju sedam tipova plagijata: potpuni plagijat, plagijat prijevodom, *copy/paste* plagijat, parafraziranje, autoplagijat, *ghostwriter* te citiranje izvan konteksta (Schwarzenegger i Wohlers, 2006).

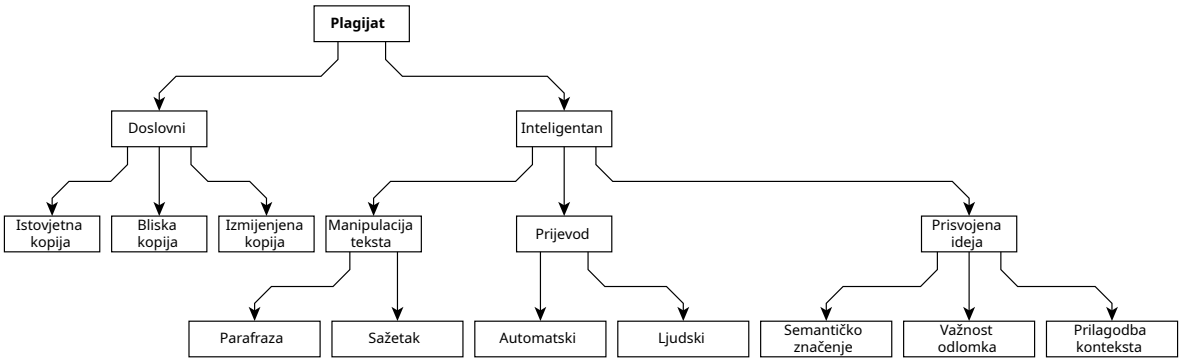
Roig (2006) razlikuje dva temeljna tipa akademskog plagiranja: plagiranje ideja i plagiranje teksta (Roig, 2006). Potonje dalje vrlo detaljno raščlanjuje na: doslovno plagiranje (engl. *verbatim*), mozaik (engl. *patchwriting* and *paraphragiarism*), nepravilno parafraziranje (engl. *inappropriate paraphrasing*), parafraziranje i sažimanje tuđeg rada (engl. *paraphrasing and summarizing of others' work*), autoplagiranje (engl. *self-plagiarism*), dupliciranje ili objava redundantnih publikacija (engl. *duplicate and redundant publication*), usložnjavanje ili fragmentiranje podataka (engl. *data augmentation or fragmentation*), nepravilno korištenje referencija (engl. i*nappropriate manipulation of references*), pretjerano citiranje (engl. *citation stuffing*), citiranje izvora koji nisu pročitani ili potpuno shvaćeni (engl. *citing sources that were not read or thoroughly understood*), minorizacija tuđih izvora (engl. *reduced recognition of borrowing*), selektivno izvještavanje o korištenoj literaturi (engl. *selective reporting of literature*), selektivno izvještavanje o korištenoj metodologiji (engl. *selective reporting of methodology*), selektivno izvještavanje o rezultatima (engl. *selective reporting of results*) i dogovorno tuđe autorstvo (engl. *ghost authorship*). Pored imenovanih tipova Roig (2026) navodi i 27 detaljnih smjernica za izbjegavanje plagiranja.

Joy i sur. (2009) promišljaju taksonomiju plagijata unutar četiri međusobno komplementarna aspekta: izvor plagiranja, način plagiranja, objekt plagiranja te ekstrinzični aspekt plagiranja. Rezultat takva promišljanja autora jest taksonomija sa 6 kategorija (plagiranje i kopiranje, referenciranje, varanje i neprimjerena suradnja, etičnost i posljedice, plagiranje izvornoga programskog koda, plagiranje dokumentacije izvornoga programskog koda) i čak 23 potkategorije plagiranja (Joy i sur., 2009).

Kakkonen i Mozgovoy (2010) razvijaju poprilično drukčiju podjelu: doslovna kopija, plagijat parafraziranjem, tehnički prikriveni plagijat, namjerno netočno korištenje literature i

teški plagijat, pri čemu u posljednju kategoriju uključuju (a) korištenje tuđih ideja, koncepata i mišljenja, (b) prijevod, (c) tuđe autorstvo (engl. *ghostwriting*) i (d) umjetnički plagijat (Kakkonen i Mozgovoy, 2010).

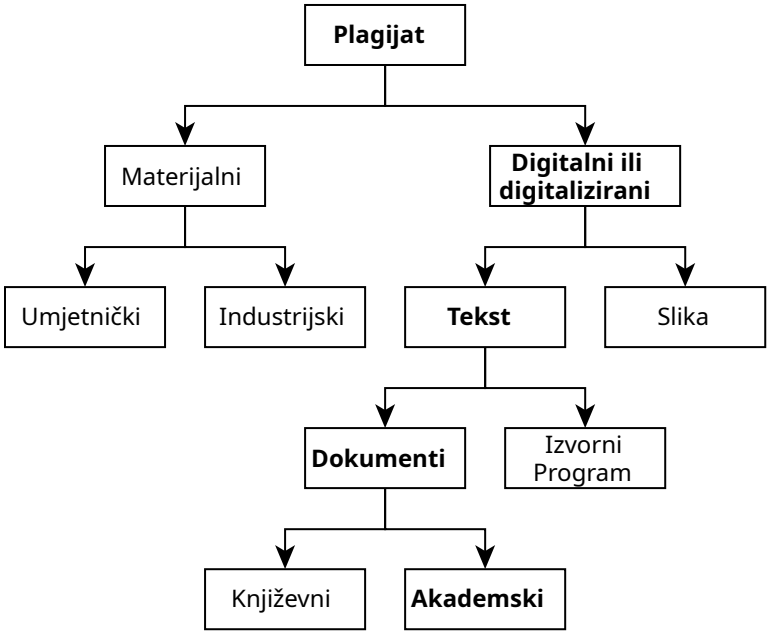
Alzahrani i sur. (2012) predlažu taksonomiju plagiranja prikazanu na slici 1. Koncept na kojem je zasnovana njihova taksonomija jest ponašanje autora prilikom plagiranja, odnosno način plagiranja (Alzahrani i sur., 2012).



Slika 1. Taksonomija plagiranja prema: Alzahrani i sur. (2012)

Plagijate možemo dijeliti na tipove ili vrste prema više kriterija. **Prema namjeri plagijatora**, postoji trivijalna podjela na namjerne i nenamjerne plagijate, no od takve podjele nema ni praktične ni teorijske koristi, osim što ona može imati različite posljedice za otkrivenog plagijatora.

Prema porijeklu i namjeni (slika 2), objekti plagiranja mogu biti materijalni (industrijski, umjetnički) i nematerijalni. Nematerijalni mogu biti u izvorno digitalnom obliku (tekstovi, izvorni programi i sl.) ili se mogu digitalizirati (umjetničke slike, pjesme i sl.).



Slika 2. Objekti plagiranja

Tekstne plagijate možemo dijeliti na akademske i književne (Meuschke i Gipp, 2013). Književni plagijati nanose umjetničku i financijsku štetu izvornom autoru. Akademski plagijati mogu izazvati akademsku i posrednu financijsku štetu. U akademskim se krugovima stoga provodi sustavna provjera tekstnih dokumenata i sustavna borba protiv plagiranja.

Akademski plagijati mogu se, prema **kriteriju tehničke realizacije plagiranja**, podijeliti na sljedeće vrste (Beames, 2012; Juričić, 2012):

- **Klon** ili potpuni plagijat (engl. *clon*) – podmetanje tuđega dokumenta kao svojega
- **Prijevod** (engl. *translation*) – prijevod tuđega dokumenta s drugog jezika bez navođenja autorstva i dozvole autora
- **Kopija** (engl. *copy*) – dokument koji sadrži znatan udio teksta iz jednoga izvora bez značajnije promjene
- **Supstitut** (engl. *find*/*replace*) – originalnom su dokumentu zamijenjene ključne riječi i izričaji, ali je dokument zadržao prvobitni smisao i sadržaj izvornoga dokumenta
- **Spoj** (engl. *remix*) – dokument u kome su parafrazirani drugi dokumenti, a spojeni su na način da djeluju kao smislena cjelina
- **Autoplagijat** (engl. *recycle*) – korištenje vlastitih ranijih dokumenata bez odgovarajuće naznake

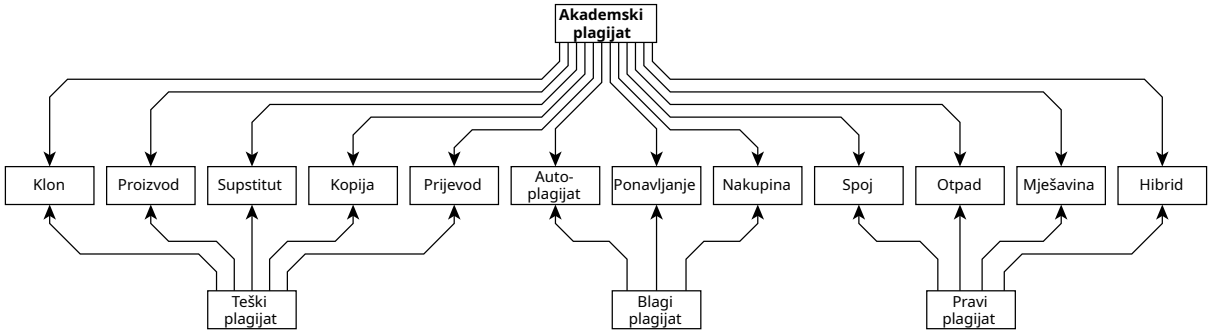
- **Hibrid** (engl. *hybrid*) – dokument u kome su kombinirani korektno citirani i kopirani dijelovi
- **Mješavina** (engl. *mashup*) – nekonzistentna mješavina dokumenata različitih izvora bez korektnog citiranja
- **Otpad** (engl. *error*) – dokument koji uključuje citate iz nepostojećih ili netočnih izvora
- **Nakupina** (engl. *aggregator*) – dokument u kome se pravilno citiraju izvori, ali ne sadrži originalnost
- **Ponavljanje** (engl. *re-tweet*) – dokument koji uključuje odgovarajuće citate, ali se previše veže na tekst ili strukturu izvornih dokumenata
- **Proizvod** (engl. *ghostwriter*) – dokument koji je potpisani autor zapravo pribavio od nekoga drugog autora kao (najčešće plaćenu) uslugu.

Prethodna podjela na 12 tipova mogla bi se, prema **kriteriju potencijalne težine posljedica**, reducirati na tri kategorije (Vrbanec i Meštrović, 2021b). **Teški plagijat** obuhvaća sljedeće tipove: klon, prijevod, kopija, supstitut i proizvod. U njima su i namjera i potencijalna šteta od plagiranja najveći, a plagijator najbezobzirniji ili najnaivniji. **Pravi plagijat** obuhvaća sljedeće tipove: spoj, hibrid, mješavina i otpad. U akademskoj su zajednici takvi pokušaji plagiranja učestali, posebice kod realizacije studentskih obaveza. Teško je razlučiti namjeru, neznanje ili naivnost autora plagijata, a teško je i njihovo otkrivanje. **Blagi plagijat** obuhvaća sljedeće tipove: autoplagijat, nakupina i ponavljanje. S moralnoga, etičkog i pravnog stajališta ta je kategorija plagijata najbenignija, što je nikako ne čini i dopuštenom ili opravdanom.

Te dvije podjele nisu međusobno potpuno neovisne. To je lakše vidjeti ako uvedemo još jedan pragmatični kriterij podjele: mogućnost automatiziranoga otkrivanja, tj. **kriterij složenosti otkrivanja**. Dakle, unutar podjele po tipu, sve tipove plagijata možemo podijeliti na one koji se **lako** ili **teško otkrivaju**, što rezultira matricom prikazanom tablicom 1 i slikom 3 (Vrbanec i Meštrović, 2021b). Lako otkrivanje podrazumijeva da ih je moguće otkriti automatiziranim programskim sustavima za provjeru plagijata (ili utvrđivanja originalnosti), a složeno otkrivanje podrazumijeva da je za njihovo otkrivanje potrebna analiza ljudskog eksperta.

Tablica 1. Matrica kategorije/složenost otkrivanja plagijata

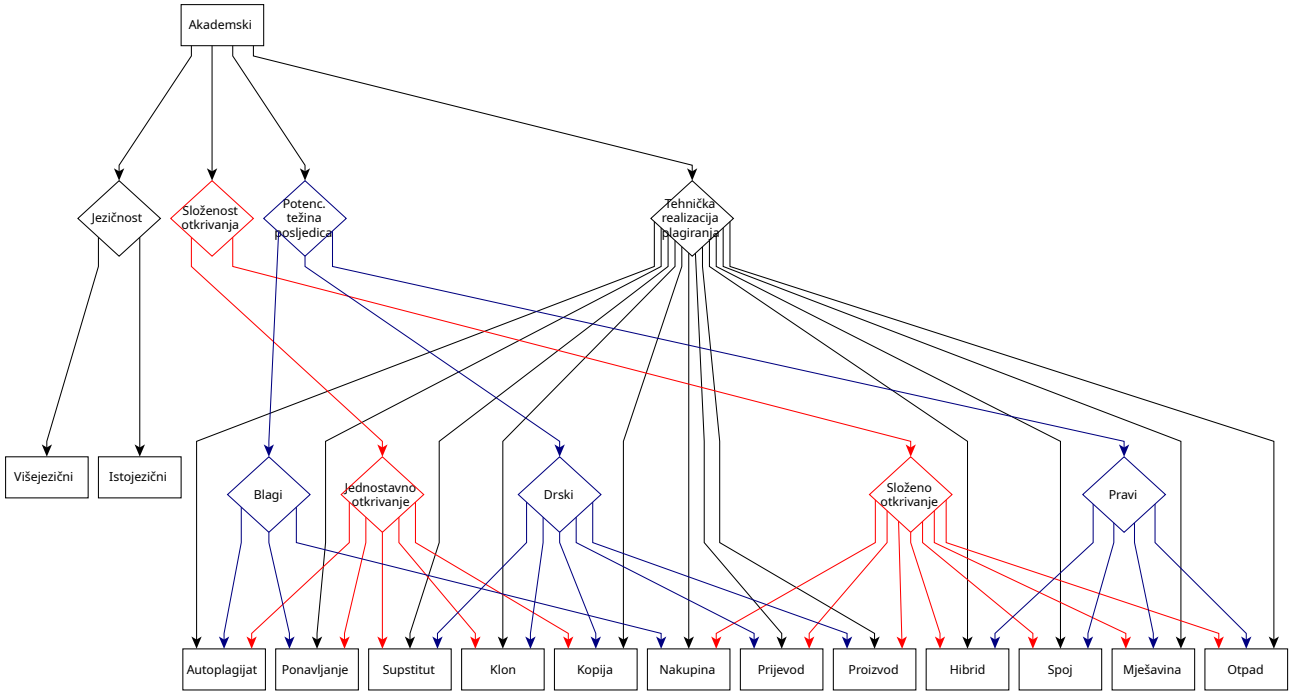
Kategorija	**Složenost otkrivanja**	
	Jednostavno	*Složeno*
Teški	klon, kopija, supstitut	prijevod, proizvod
Pravi	-	spoj, hibrid, mješavina, otpad
Blagi	autoplagijat, ponavljanje	nakupina



Slika 3. Tipovi akademskih plagijata

Prema jezičnom porijeklu, plagijate je moguće podijeliti na **istojezične** i plagijate **prijevodom**. Plagijati prijevodom mogu nastati plagiranjem dokumenata s jednog ili između više jezika, a preduvjet za njihovo programsko otkrivanje jest pristup programskim sustavima za automatsko prevođenje.

Cijela se taksonomija akademskih plagijata može prikazati dijagramom na slici 4, a prema navedenim četirima kriterijima: (više)jezičnost, složenost otkrivanja, težina potencijalnih posljedica i tehnička realizacija.



Slika 4. Taksonomija akademskih plagijata

2.4. Metode akademskog plagiranja

Prema Alzahraniju i suradnicima (2012), plagirani dijelovi mogu nastati parafraziranjem, sažimanjem originalnoga teksta, kombiniranjem, restrukturiranjem, generalizacijom ili specifikacijom koncepata (Alzahrani i sur., 2012). Preduvjet za sumnju u plagiranje teksta jest to da izvori nisu (ispravno) referencirani.

Maurer i sur. (2006) identificiraju različite metode plagiranja, uključujući: kopiranje i lijepljenje teksta, plagiranje ideja koje nisu općeprihvaćene, parafraziranje tuđih misli uz gramatičke preinake, umjetničko plagiranje kroz korištenje drugih medija za istu ideju, plagiranje programskoga koda bez dopuštenja ili navođenja izvora, izostavljanje ili nepotpuno navođenje izvora, netočno korištenje navodnika, dezinformiranje referencijama koje vode na krive ili nepostojeće izvore te plagiranje prijevodom bez referencije (Maurer i sur., 2006).

Turnitin, vodeći proizvođač softvera za detekciju plagijata, odnosno utvrđivanje originalnosti dokumenata, razlikuje metode plagiranja u akademskom kontekstu i u istraživačkim radovima (Turnitin Europe, 2016): akademsko plagiranje obuhvaća predaju tuđega rada kao vlastitoga, kopiranje riječi ili ideja bez pripisivanja zasluga izvornom autoru, preuzimanje većine riječi i ideja koje ugrožavaju originalnost rada, ponovnu predaju istoga

rada, nekorištenje navodnika prilikom citiranja, davanje netočnih informacija o izvorima, korištenje tuđih rečenica uz zamjenu pojedinih riječi te korištenje tuđih ideja bez odgovarajućega referenciranja. Nameće se zaključak da *Turnitin* doista deklarira akademsko plagiranje kao plagiranje koje se događa u sklopu srednjoškolskoga, prijediplomskoga i diplomskoga formalnog obrazovanja, u okolnostima ispunjavanja obaveze pisanja seminara i drugih radova, gdje se i ne očekuje znatniji istraživački napor od učenika i studenata. S druge strane, *Turnitin* definira specifične metode plagiranja na znanstvenoistraživačkoj razini, gdje se od radova očekuje originalnost, inovativnost i doprinos razvoju znanosti ili struke (Turnitin Europe, 2016). Te metode uključuju objavljivanje radova o tuđim istraživanjima ili prisvajanje tuđih istraživanja, navođenje nekorištenih izvora, ponovno korištenje istraživanja ili radova bez odgovarajućega referenciranja, parafraziranje tuđih radova i predstavljanje takvih radova kao vlastitih, ponavljanje podataka ili teksta iz sličnih istraživanja bez pravilnoga navođenja izvora, slanje istih radova u više publikacija te propuste u referenciranju citata ili nepriznavanje doprinosa suradnika. Razdvajanje slučajeva i metoda plagiranja na dvije razine (obrazovna i akademska) djeluje pretjerano s obzirom na to da je plagiranje uvijek plagiranje bez obzira na kontekst i obrazovnu razinu aktera plagiranja. Te dvije skupine metoda plagiranja nepotrebno su i umjetno razdvojene te bi ih trebalo objediniti u istu skupinu metoda akademskoga plagiranja (Vrbanec i Meštrović, 2021b).

Tradicionalne metode plagiranja od 2018. godine dopunjene su i unaprijeđene mogućnostima koje pružaju jezični modeli (engl. *Language Models*, LM): generiranje i preformuliranje teksta, automatsko parafraziranje i kompiliranje iz nedeklarirane kombinacije izvora. Jezični modeli su klasa algoritama za obradu prirodnoga jezika koji se bave predviđanjem sljedeće riječi u nizu teksta (Nandakumar i sur., 2023; Radford i sur., 2019). Generativna umjetna inteligencija, a posebno generativni (veliki) jezični modeli, vrsta je umjetne inteligencije koja stvara novi sadržaj, poput teksta, slika ili glazbe; to je skup tehnika umjetne inteligencije i modela dizajniranih za učenje inherentnih obrazaca skupa podataka te njihove strukture, a služe za generiranje novih, najvjerojatnijih podataka iz izvornoga skupa podataka (Pinaya i sur., 2023). Modeli uče iz golemih količina podataka kako bi proizveli originalna djela. Primjenjuju se u pisanju, umjetnosti, glazbi, dizajnu, obrazovanju i zabavi. Veliki jezični modeli imaju i određenih ograničenja, odnosno probleme poput pristranosti, halucinacija, kvalitete i etičnosti korištenja generiranih sadržaja. Generiranje teksta pomoću tih modela može se koristiti za stvaranje prikrivenih plagijata, što dodatno otežava njihovo

otkrivanje. S obzirom na to da generativni modeli uče iz postojećega sadržaja, postoji rizik da generirani tekstovi budu vrlo slični izvornim radovima, ali dovoljno promijenjeni da je teško ili nemoguće njihovo otkrivanje tradicionalnim alatima za detekciju plagijata. Uz mogućnost generiranja sadržaja koji se malo razlikuju od originala, generativni modeli olakšavaju proizvodnju sadržaja, što povećava količinu potencijalnih plagijata. Zbog svega toga unaprjeđuju se i razvijaju alati za provjeru originalnosti koji su sposobni otkrivati suptilne varijacije (T. Williams, 2023), ali se ipak još ne može govoriti o otkrivanju na semantičkoj razini jer se ti alati temelje na: (a) statističkom utvrđivanju učestalosti korištenja riječi koje su statistički zastupljenije kod teksta generiranoga generativnim jezičnim modelima nego što je to u uobičajenome ljudskom govoru i pismu te (b) na prepoznavanju podjednake veličine odlomaka teksta, što nije uobičajeno kod ljudskih autora.

2.5. Klasifikacija i pristupi otkrivanju plagiranja

Više je pristupa otkivanju plagiranja, a uglavnom se temelje na korpusima, znanju ili na njihovoj kombinaciji. Lancaster i Culwin (2005) klasificirali su pristupe otkrivanju plagijata prema pet kriterija (Lancaster i Culwin, 2005). Prema **tradicionalnoj klasifikaciji**, dokumentima se računaju ili svojstva (engl. *Attribute Counting Systems*) ili struktura (engl. *Structure Metric Systems*). Lancaster i Culwin (2005) smatraju takvu klasifikaciju nedorečenom jer neki sustavi imaju pristup koji ne pripada nijednoj od dviju klasa. Prema **tipu korpusa** dokumenata koji se obrađuje, autori predlažu više podjela. Tako, prema vrsti dokumenata, korpus može sadržavati izvorni tekst programa, tekstne dokumente ili kombinaciju jednih i drugih. S obzirom na izvor dokumenata, korpusi mogu biti interni (dostupni samo unutar organizacije), eksterni (prikupljeni s interneta) ili mješoviti (kombinacija internih i eksternih izvora). Nadalje, alati se mogu razlikovati prema načinu rada, odnosno primjenjuju li tokenizaciju prilikom analize teksta. Prema **dostupnosti sustava za otkrivanje plagijata**, mogu se dalje klasificirati s obzirom na smještaj i otvorenost. S obzirom na smještaj, alati mogu biti lokalni (instalirani na računalu korisnika) ili dostupni na *webu* (pristupačni putem internetskog preglednika). Nadalje, prema otvorenosti, alati mogu biti javni (dostupni svima) ili privatni (ograničeni na određenu skupinu korisnika). Klasifikacija pristupa za otkrivanje plagijata može se temeljiti i **prema broju dokumenata** koje istodobno obrađuje korištena metrika. Metrike mogu biti singularne (analiziraju jedan

dokument), parne (uspoređuju dva dokumenta) ili korpusne (analiziraju veći broj dokumenata istodobno). Klasifikacija pristupa za otkrivanje plagijata može se provesti i **prema složenosti korištenih metrika**. Metrike za otkrivanje sličnosti tekstova mogu biti površinske (analiziraju samo tekstni sadržaj) ili strukturne (uzimaju u obzir i strukturu dokumenta).

Maurer i suradnici (2006) predlažu strategiju otkrivanja plagijata koja se sastoji od tri ključne faze. Prva faza uključuje korištenje lokalnog repozitorija dokumenata gdje se provjeravani dokument uspoređuje riječ po riječ s potencijalnim izvorima plagijata. U drugoj fazi dokument se uspoređuje sa svim dostupnim *web*-izvorima, ali fokus je na usporedbi karakterističnih dijelova ili rečenica, a ne cijelih dokumenata. Treća faza koristi stilometrijski algoritam za jezičnu analizu koji analizira stil uzastopnih odlomaka unutar dokumenta. Nedosljednosti ili promjene stila mogu signalizirati potencijalno plagiranje (Maurer i sur., 2006).

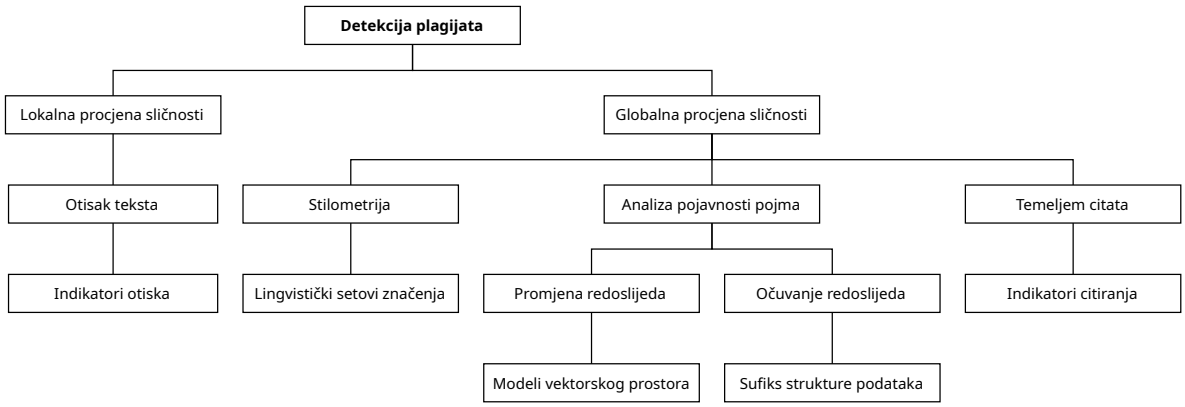
Culwin i Lancaster (2001b) predlažu četverofazni model otkrivanja plagijata. Prva faza obuhvaća prikupljanje relevantnih dokumenata i njihovo pohranjivanje u repozitorij. U drugoj, detekcijskoj fazi, programski sustav identificira sumnjive parove dokumenata koji bi mogli sadržavati plagijat. Treća faza, faza potvrde, uključuje ljudskog eksperta koji analizira sumnjive parove i potvrđuje ili odbacuje sumnju u plagijat. U posljednjoj, istraživačkoj fazi ljudski ekspert detaljno analizira potvrđene slučajeve plagijata i određuje odgovarajuće sankcije za plagijatora (Culwin i Lancaster, 2001b).

Williams (2005) predlaže evolucijski pristup borbi protiv plagiranja koji se sastoji od tri strategije. Prva strategija uključuje jednostavne tehnike pretraživanja *weba* koje koriste pojedini predavači. Druga strategija podrazumijeva korištenje besplatnih programa za otkrivanje plagijata unutar korpusa studentskih radova. Treća, najnaprednija strategija obuhvaća korištenje složenih sustavnih pristupa, uključujući angažman komercijalnih agencija specijaliziranih za detekciju plagijata (J. B. Williams, 2005).

Metode otkrivanja plagiranja temelje se na pripadnim algoritmima, često i više njih, te na heuristici. Heuristika obično nastaje na temelju iskustva autora metode, a i ona se, ako je moguće, formalizira i pretvara u algoritamski oblik kako bi se mogla programski primijeniti te kako bi se objasnila njezina logika, ispravnost rada i dokazala njezina formalna utemeljenost u stručnim i znanstvenim krugovima. Idealan algoritam za otkrivanje plagijata trebao bi biti sposoban detektirati različite oblike nedopuštenoga korištenja tuđega rada (Culwin i Lancaster, 2001a; Kakkonen i Mozgovoy, 2010; Lancaster i Culwin, 2005; Maurer i sur.,

2006; J. B. Williams, 2005). To uključuje doslovno kopiranje izvorno digitalnih ili digitaliziranih analognih izvora, kao i različite oblike parafraziranja poput dodavanja ili uklanjanja riječi, namjernih pravopisnih pogrešaka, zamjene riječi sinonimima ili promjene redoslijeda riječi. Algoritam bi trebao biti otporan na tehničke trikove kojima se pokušavaju zaobići postojeći sustavi, poput korištenja sličnih fontova, nevidljivih znakova ili slika teksta umjesto samoga teksta. Također, trebao bi prepoznati namjerno pogrešno referiranje, uključujući netočno korištenje navodnika, nepostojeće referencije ili poveznice. Uz to idealan algoritam trebao bi biti sposoban detektirati i teže oblike plagijata poput plagiranja ideja, prijevoda, korištenja teksta skrivenog pisca ili umjetničkog plagijata, gdje se tuđi rad predstavlja na drugom mediju. Takav idealan algoritam za otkrivanje plagijata, naravno, ne postoji, a s obzirom na složenost, više je algoritama potrebno za obavljanje ukupnog zadatka otkrivanja plagijata. To je složen problem koji do danas nije riješen, no razvijali su se mnogobrojni pristupi njegovu rješavanju. Prvi su sustavi koristili metode analize teksta poput podudaranja nizova znakova (engl. *string*) i statističke analize pojavnosti riječi i izraza. Potom su se razvijale metode uzimanja neke vrste otisaka i metode stilometrijske analize. Navedene su metode tijekom vremena postale vrlo sofisticirane, no ne i dostatne.

Prepoznavanje plagiranja u početku je bilo u domeni statističke analize teksta, danas je prvenstveno dio obrade prirodnoga jezika (engl. *Natural Language Processing*, NLP), a od 2013. godine intenzivno se uključuje i strojno učenje (engl. *Machine Learning*, ML). Meuschke i Gipp (2013) predložili su moguću klasifikaciju metoda za otkrivanje plagijata prikazanu slikom 5 (Meuschke i Gipp, 2013). Prema prikazanoj klasifikaciji, metode procjene lokalne sličnosti uspoređuju ograničene dijelove teksta i pri tome se najčešće koristi neka vrsta uzimanja otisaka (engl. *fingerprinting*). Metode procjene globalne sličnosti procjenjuju veće dijelove teksta ili cijele dokumente i te metode najčešće analiziraju pojavnosti pojmova, koriste uzorke citata ili stilometrijske metode kojima traže razlike unutar dokumenta.



Slika 5. Klasifikacija metoda detekcije plagijata

2.5.1. Ekstrinzične i intrinzične metode

Postojeće metode otkrivanja sličnosti dokumenata možemo klasificirati u vanjske (ili ekstrinzične) te unutarnje (ili intrinzične), prema tome traži li se plagiranje uspoređujući dokumente (iz korpusa dokumenata) ili se unutar samoga dokumenta traže dokazi plagiranja (M. Chong i sur., 2010; Gipp i sur., 2011).

Ekstrinzične metode fokusiraju se na usporedbu teksta s vanjskim izvorima kako bi se identificirali potencijalno plagirani dijelovi. Ti vanjski izvori mogu biti raznoliki, uključujući znanstvene baze podataka kao što su *Web of Science* ili *Scopus*, opće pretraživače poput *Googlea* ili specijalizirane poput *Google Scholara* ili *Semantic Scholara*, *web*-stranice, knjige te digitalne repozitorije akademskih radova. Specijalizirani softveri za otkrivanje plagijata koriste napredne algoritme za usporedbu tekstova s velikim brojem vanjskih izvora. **Intrinzične metode**, s druge strane, analiziraju tekst bez oslanjanja na vanjske izvore. Te se metode usredotočuju na: a) identifikaciju promjena u stilu pisanja, vokabularu, strukturi rečenica, korištenju interpunkcijskih znakova, stilu citiranja, b) analizu učestalosti riječi, c) analizu dužine rečenica, d) analizu sintakse i semantike teksta te druge jezične značajke koje mogu upozoravati na plagijat. Primjer intrinzične metode je stilometrijska analiza teksta, tj. *stilometrija*, pri kojoj se traže odstupanja u stilu pisanja unutar istoga dokumenta, što može biti indikator plagiranja (Oberreuter i Velásquez, 2013), budući da svaka osoba ima svoj jedinstven stil pisanja (Maurer i sur., 2006).

Stilometrija u svojem naprednijem obliku može biti i **kombinirano intrinzično-ekstrinzična** metoda jer se kvantifikacija parametara stila može izvući na intrinzičan način (iz

samoga dokumenta) ili ekstrinzičan način (iz drugih radova istoga autora). Eissen i Stein razlikuju pet kategorija stilometrijskih parametara (Zu Eissen i Stein, 2006): (i) statistiku teksta koja djeluje na razini znakova, (ii) sintaktičke značajke koje mjere stil pisanja na rečeničnoj razini, (iii) značajke dijelova teksta koje kvantificiraju korištenje klasa riječi, (iv) zatvorene klase setova riječi koje broje posebne riječi i (v) strukturne značajke koje odražavaju organizaciju teksta.

Intrinzičnim su metodama svojstvene tri sistemske slabosti. Prva je u tome što tekst koji se provjerava mora biti od jednog autora. Druga je slabost u tome što provjeravani tekst mora biti značajnije veličine (kao jedinstveno veće djelo ili kao skupina manjih djela jednog autora) da bi se mogle utvrditi autorove osobitosti. Treća je slabost nedostatak potvrde ili dokaza da dio dokumenta nije originalan, s obzirom na to da intrinzične metode ne koriste korpus dokumenata za usporedbu.

2.5.2. Statističke metode/mjere

Statističke metode iz dokumenata izvlače frekvencije riječi, računaju njihove težinske vrijednosti (pondere) te računaju statističke mjere udaljenosti (Ushio i Liberatore, 2024). Statističke su metode često sastavnice drugih metoda. Tako se primjerice često koriste u kombinaciji s algoritmima strojnog učenja (Ushio i Liberatore, 2024).

Jaccardova sličnost je mjera koja se računa kao omjer veličine presjeka skupova (riječi) i njihove unije (Jaccard, 1912). Često se koristi za mjerenje sličnosti između dva skupa (riječi) u tekstovima. Koristi se u analizi podataka, strojnom učenju te informacijskoj tehnologiji za kvantificiranje sličnosti između dvaju skupova podataka ili atributa. Za dva skupa A i B Jaccardova sličnost definirana je kao:

$$S_{Jaccard}(A,B)=\frac{|A\cap B|}{|A\cup B|} \quad (1)$$

gdje |A∩B| predstavlja broj elemenata (riječi) u presjeku skupova A i B, a |A∪B| predstavlja broj elemenata (riječi) u uniji skupova A i B.

Kulback-Leiblerova divergencija je mjera koja kvantificira razliku između dviju distribucija vjerojatnosti P i Q (Cover, 1999). U kontekstu analize teksta ta se mjera može koristiti za usporedbu distribucije riječi u dva teksta. Kullback-Leiblerova divergencija izražava koliko se informacija gubi kada se jedna distribucija koristi za aproksimaciju druge. Koristi se u statistici, teoriji informacija, strojnom učenju te dubokom učenju. Za dvije diskretne vjerojatnosne distribucije P i Q, Kullback-Leiblerova divergencija definirana je kao:

$$D_{KLD}(P\|Q)=\sum_i P(i)\log\frac{P(i)}{Q(i)} \tag{2}$$

gdje je *i* indeks koji prolazi kroz sve moguće ishode distribucija koje su definirane vjerojatnostima u distribucijama *P* i *Q*. U diskretnom slučaju to su pojedinačni ishodi, a u kontinuiranom slučaju to su vrijednosti kontinuirane varijable. Kulback-Leiblerova divergencija može se u literaturi naći i pod drugim pojmovima (Cover, 1999), npr. relativna entropija (engl. *Relative Entropy*) ili informacijska divergencija (engl. *Information Divergence*).

Chi-kvadrat test sličnosti je statistički test koji uspoređuje očekivane i stvarne frekvencije pojave riječi u tekstovima i često se koristi za analizu sličnosti između distribucija riječi u tekstovima (Deisenroth i sur., 2020). Mjera sličnosti temelji se na tome koliko se frekvencije u skupovima podataka razlikuju od onih koje bismo očekivali da su dva skupa identična. Chi-kvadrat test se često koristi u analizi podataka, statistici, i istraživačkim radovima za provjeru hipoteza. Za dvije distribucije O (promatrane frekvencije) i E (očekivane frekvencije), Chi-kvadrat test sličnosti χ^2 definiran je kao

$$\chi^2=\sum_{i=1}^{n}\frac{(O_i-E_i)^2}{E_i} \tag{3}$$

gdje *Oi* predstavlja promatranu frekvenciju u kategoriji *i*, dok *Ei* predstavlja očekivanu frekvenciju u kategoriji *i*.

Neki autori u statističke veličine ubrajaju i različite mjere udaljenosti (M. Li i sur., 2004): Hammingovu udaljenost, Euklidovu udaljenost, Lempel-Zivovu udaljenost, kompresijsku udaljenost, informacijsku udaljenost i normaliziranu informacijsku udaljenost. K-znakovna statistika poprilično je učinkovita; primjerice dvoznakovna pouzdano identificira jezik kojim je dokument pisan, a troznakovna klasificira dokument.

kod statističkih metoda postoje dva pristupa (Leacock i sur., 1998): (a) pristup temeljen na rječnicima (Leacock i sur., 1998; Z. Wu i Palmer, 1994) koji se oslanja na

definicije i (b) pristup temeljen na korpusima (P. Turney, 2001) koji koristi pojavnosti istih riječi unutar velikoga korpusa tekstova.

2.5.3. Geometrijske ili strukturne mjere

Geometrijske ili strukturne mjere obično se temelje na geometrijskim ili strukturnim svojstvima vektorskoga prostora u koji se preslikavaju tekstovi (Levy i sur., 2024). Za te mjere sličnosti, koje nisu statističke, ponekad se koristi i naziv *neparametarske mjere* jer one ne zahtijevaju pretpostavke o parametrima statističke distribucije i često se temelje na strukturi podataka. Neke su od njih kosinusna sličnost, te Euklidska udaljenost, Mahalanobisova udaljenost, Hellingerova udaljenost, udaljenost Minkowskoga i *Word Moving* udaljenost.

Kosinusna sličnost je kosinus kuta između dvaju vektora koji predstavljaju tekstove u višedimenzionalnom vektorskom prostoru, mjera koja se često koristi u analizi teksta i obradi prirodnoga jezika (Manning i sur., 2009), a i u ovome istraživanju potvrdila se kao primjerena, direktna, jednostavna i računalno nezahtjevna. Kosinusna sličnost mjeri kosinus kuta između dva vektora u n-dimenzionalnom prostoru, tj. između vektora X i Y definirana je kao

$$S_{Cosine} = \cos(\theta) = \frac{X \cdot Y}{|X| \cdot |Y|} \tag{4}$$

gdje je X·Y skalarni produkt vektora X i Y, odnosno zbroj umnožaka odgovarajućih komponenti dvaju vektora, a ||X|| i ||Y|| su duljine vektora X i Y.

Meka kosinusna sličnost (engl. *soft cosine similarity*) proširuje klasičnu kosinusnu sličnost uzimajući u obzir semantičku sličnost između komponenti vektora. To znači da ne uspoređuje samo kut između vektora već i sličnost između njihovih pojedinačnih dimenzija. To proširenje omogućuje bolju usporedbu tekstova jer se ne gleda samo egzaktno podudaranje riječi već i njihova sličnost. Formalno, meka kosinusna sličnost definirana je kao

$$S_{SoftCosine} = \frac{X^T S Y}{\sqrt{X^T S X} \cdot \sqrt{Y^T S Y}} \tag{5}$$

gdje je S matrica sličnosti između pojedinačnih dimenzija vektora, $X^T S Y$ jje generalizirani skalarni produkt koji uzima u obzir sličnost između komponenti, $\|X\|_S = \sqrt{X^T S X}$ i $\|Y\|_S = \sqrt{Y^T S Y}$ su normirane duljine vektora s obzirom na matricu sličnosti S. Matrica sličnosti S može se dobiti na tri načina: pomoću vektorskih reprezentacija riječi, pomoću

matrica supojavljivanja (engl. *co-occurence*)[1], te pomoću leksičkih resursa[2] poput *WordNeta* ili *BabelNeta*. U istraživanju je korištena *Python* biblioteka otvorenoga koda *Gensim*, koja u svrhu stvaranja matrice *S* potrebne za meku kosinusnu sličnost koristi ugrađene *word embeddings* modele za dobivanje vektora riječi i izračunavanje kosinusne sličnosti među njima.

Euklidska udaljenost je mjera udaljenosti između dvije točke u Euklidskom prostoru, a definirana je kao duljina pravca koji povezuje te dvije točke (Deisenroth i sur., 2020).

$$D_{Euclid}(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{6}$$

gdje su $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ točke zadane svojim n-dimenzionalnim koordinatama.

Mahalanobisova udaljenost je mjera udaljenosti između točke i distribucije ili između dviju točaka u višedimenzionalnom prostoru, koja uzima u obzir korelacije između varijabli (McLachlan, 1999). Koristi se za identificiranje anomalija i u statističkoj analizi podataka, a definira se kao

$$D_{Mahalanobis}(X,Y)=\sqrt{(X-Y)^T S^{-1}(X-Y)} \tag{7}$$

gdje su $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ višedimenzionalni vektori, *S* je kovarijancijska[3] matrica varijabli, a $(X-Y)^T$ je transponirani vektor razlike vektora *X* i *Y*.

Hellingerova udaljenost je mjera udaljenosti između dviju distribucija vjerojatnosti, često korištena za usporedbu distribucija riječi u tekstovima (Gibbs i Su, 2002). Prema tome radi li se o diskretnim ili kontinuiranim distribucijama, računa se na dva moguća načina,

$$H(P,Q)=\frac{1}{\sqrt{2}}\cdot\sqrt{\sum_{i=1}^{n}(\sqrt{p_i}-\sqrt{q_i})^2} \tag{8}$$

za diskretne distribucije i

$$H(P,Q)=\frac{1}{\sqrt{2}}\cdot\sqrt{\int(\sqrt{p(x)}-\sqrt{q(x)})^2\,dx} \tag{9}$$

za kontinuirane distribucije,

pri čemu su *P* i *Q* dvije diskretne vjerojatnosne distribucije $P=(p_1, p_2, ..., p_n)$ i $Q=(q_1, q_2, ..., q_n)$. Hellingerova udaljenost ima vrijednosti između 0 i 1.

1 Matrica sufrekvencija bilježi koliko se riječi često pojavljuju zajedno u istom kontekstu (npr. prozorskom kontekstu od 5 riječi). Normalizacijom matrice može se dobiti mjera sličnosti između riječi.

2 Koristeći *WordNet* ili *BabelNet* može se definirati sličnost na temelju hijerarhije sinonima i hiperonima (npr. pomoću Leacock-Chodorowe ili Wu-Palmerove sličnosti).

3 Kovarijancijska matrica je kvadratna matrica koja sadrži kovarijance između svih parova varijabli u skupu podataka. Kovarijanca između dvije varijabli *X* i *Y* mjeri koliko se te varijable zajedno mijenjaju.

Manhattan udaljenost, poznata i kao *taksi udaljenost*, mjera je udaljenosti između dviju točaka u prostoru koja se računa kao zbroj apsolutnih razlika njihovih koordinata (Tan i sur., 2014). Naziv *Manhattan* dolazi iz geometrije grada poput ulica na Manhattanu, gdje se kretanje između dviju točaka vrši prema pravokutnim linijama, a izračunava se kao

$$D_{Manhattan}(X,Y)=\sum_{i=1}^{n}|x_i-y_i| \tag{10}$$

gdje su X i Y dvije točke $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ u n-dimenzionalnom prostoru.

Chebyshevljeva udaljenost ili maksimalna udaljenost je mjera udaljenosti između dviju točaka u prostoru koja se računa kao najveća apsolutna razlika između odgovarajućih koordinata točaka (Deisenroth i sur., 2020). Drugim riječima, Chebyshevljeva udaljenost definira korake koji su potrebni za pomicanje s jedne točke na drugu ako je moguće kretanje samo duž koordinatnih osi.

$$D_{Chebyshev}(X,Y)=\max_{i=1}^{n}|x_i-y_i| \tag{11}$$

gdje su X i Y dvije točke $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ u n-dimenzionalnom prostoru.

Udaljenost Minkowskoga je generalizacija Euklidske udaljenosti, koja uključuje parametar p koji određuje različite oblike metrika u prostoru (Tan i sur., 2014; Turing.com, 2024). Udaljenost Minkowskoga između dviju točaka $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ u n-dimenzionalnom prostoru definira se kao:

$$D_{Minkowski}(X,Y)=\sqrt[p]{\sum_{i=1}^{n}|x_i-y_i|^p} \tag{12}$$

Kada je p=1, udaljenost Minkowskoga postaje Manhattan udaljenost kada je p=2, postaje Euklidska udaljenost, a kada p teži prema beskonačnosti, udaljenost Minkowskoga postaje Chebyshevljeva udaljenost (ili šahovska udaljenost), koja mjeri najveću apsolutnu razliku između koordinata. Parametar p dakle omogućuje fleksibilnost u definiranju udaljenosti pa tako udaljenost Minkowskoga može modelirati različite vrste udaljenosti.

Udaljenost premještanja riječi (engl. *Word Mover's Distance*, WMD) je mjera sličnosti između dvaju dokumenata koja koristi distribucije riječi za izračunavanje minimalne udaljenosti potrebne za transformiranje jednoga dokumenta u drugi (Kusner i sur., 2015). WMD se temelji na ideji *Earth Mover's Distance* (EMD), a koja se koristi za mjerenje udaljenosti između distribucija. WMD se temelji na vektorskim reprezentacijama riječi, modela poput *Word2Vec*, gdje su riječi predstavljene kao vektori u višedimenzionalnom

prostoru. WMD izračunava minimalni ukupni trošak potrebnog premještanja riječi iz jednoga dokumenta kako bi se prekrio drugi, uzimajući u obzir semantičku sličnost riječi, a definirana je kako slijedi.

$$WMD(D_1, D_2) = \sum_{i=1}^{n} \sum_{j=1}^{m} T_{i,j} C_{i,j} \tag{13}$$

gdje $T_{i,j}$ predstavlja količinu "masa" koju treba premjestiti s riječi *i* u dokumentu D_1 na riječ *j* u dokumentu D_2, $C_{i,j}$ je trošak premještanja riječi *i* u D_1 na riječ *j* u D_2, a što se izračunava kao Euklidska udaljenost između vektora riječi *i* i *j*.

Canberra udaljenost je mjera udaljenosti između dviju točaka koja naglašava razlike između malih i velikih vrijednosti te je definirana kao

$$D_{Canberra}(X, Y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{14}$$

gdje su *X* i *Y* dvije točke $X=(x_1, x_2, ..., x_n)$ i $Y=(y_1, y_2, ..., y_n)$ u *n*-dimenzionalnom prostoru.

Klaster analiza je skup tehnika koje grupiraju tekstove prema sličnosti na temelju različitih kriterija. Hijerarhijska klaster analiza često koristi geometrijske principe. **Koeficijent siluete** (engl. *Silhouette Coefficient*) je mjera koja procjenjuje kvalitetu klasteriranja, uzimajući u obzir sličnost unutar klastera i različitost između njih.

$$S = \frac{b - a}{max(a, b)} \tag{15}$$

Metode vektorske reprezentacije koriste jezične modele dubokog učenja poput *Word2Vec* (Mikolov i sur., 2013a), *Doc2Vec* (Le i Mikolov, 2014), *FastText* (Bojanowski i sur., 2016), *GloVe* (Pennington i sur., 2014) ili druge tehnike vektorske reprezentacije kojima se riječi ili dokumenti preslikavaju u prostor gdje je sličnost geometrijski relevantna.

2.6. Pregled i nedostaci dosadašnjih istraživanja

Istraživanja sličnosti tekstova i izvornoga programskog koda imaju već dovoljno dugu povijest da se mogu razgraničiti tri faze koje bismo mogli nazvati početnom, zrelom i semantičkom. U početnoj su postavljeni temelji, u zreloj se uvelike koriste statističke metode, a u posljednje se vrijeme uz primjenu umjetne inteligencije, neuronskih mreža, dubokog učenja i (velikih) jezičnih modela istraživači sve više bave semantičkim obilježjima teksta.

2.6.1. Počeci istraživanja

Istraživanje načina detekcije plagijata započelo je prije gotovo stotinu godina. Pojavom i širenjem računalne tehnologije naglasak je stavljen na automatsko otkrivanje plagiranja. Prema Chongu, već je 1927. godine Charles Bird „prvi istraživao primjenu statističkih metoda u otkrivanju plagijata kod višestrukog izbora odgovora" (M. Y. M. Chong, 2013). Chong dalje nastavlja da su 1960-ih godina razvijene prve metode usmjerene na otkrivanje plagiranja testova s višestrukim izborom odgovora, 1970-ih godina razvijeni su prvi sustavi za utvrđivanje plagiranja pisanoga teksta – izvornoga koda programa, a za prirodne jezike 1990-ih. Potreba za detekcijom akademskih plagijata potaknula je od ranih 1960-ih godina razvoj različitih metoda. James W. Perry i Allen Kent istraživali su upotrebu računala za analizu teksta i otkrivanje sličnosti među dokumentima. Svojom su knjigom (Perry i Kent, 1958) postavili temelje za kasniji razvoj specijaliziranih alata za detekciju plagijata. John Swets i njegovi kolege istraživali su upotrebu statističkih modela za identifikaciju sličnosti među dokumentima te naznačili da se računalni algoritmi mogu koristiti za otkrivanje potencijalnih plagijata (Swets, 1964). Prema Alzahraniju i sur. (2012) prvi radovi o plagiranju tekstova i izvornog programskog koda potječu iz 1970-ih godina. Pretežito su se bavili otkrivanjem plagijata u izvornim programima pisanima u programskim jezicima *Pascal* i *C* (Alzahrani i sur., 2012). Michael E. Lesk razvio je računalni program „sif" koji (mu) je služio za usporedbu datoteka (Lesk, 1977; Swets, 1964), a koristio je algoritme za usporedbu nizova znakova. Istraživači sa Sveučilišta Cornell objavili su „*A file comparison program*" (Miller i Myers, 1985) koji je koristio tada naprednije tehnike kao što su analiza učestalosti riječi i analiza sintakse kako bi identificirao sličnost dokumenata. Donald E. Knuth sredinom 1980-ih godina razvio je „Plague" – sustav za detekciju plagijata u studentskim radovima iz informatike (Knuth, 1989). Knuthov pristup temeljio se na analizi programskih kodova, no poslije je prilagođen i za analizu prirodnoga jezika.

1990-ih godina pojavili su se radovi u kojima su predstavljene statističke računalne metode otkrivanja kopiranja u prirodnim jezicima. Znanstvenici počinju ozbiljnije publicirati radove o akademskim plagijatima. Tako primjerice Samuelson polemizira o etičnosti i kršenju autorskih prava izdavača u slučaju autoplagiranja (Samuelson, 1994). Autori na prijelazu tisućljeća uglavnom se bave problemima pronalaženja plagijata u zatvorenim sustavima unutar akademskih ustanova i *web*-plagiranjem. U 1990-im godinama istraživači su razvili programske sustave poput MOSS-a (engl. *Measure of Software Similarity*) koji je u upotrebi

od 1994. godine (Aiken, 2022; Schleimer i sur., 2003) koji su primarno fokusirani na programski kod, ali su također otvorili vrata i za analizu podudarnosti tekstova (Maurer i sur., 2006).

U prvome desetljeću 21. stoljeća razvoj metodologija i tehnologija za detekciju plagijata značajno je napredovao. Razvijene su sofisticirane tehnike obrade prirodnog jezika (NLP) i strojnoga učenja koje su omogućile precizniju analizu i detekciju plagijata. Naprimjer algoritmi za prepoznavanje obrazaca kao što su *shingling* i *fingerprinting*[1] omogućili su analizu podudarnosti tekstova na razini rečenica i fraza (Hoad i Zobel, 2003).

Istraživači na početku tisućljeća pokušavali su sljedeće: (a) dotjerati postojeće sustave kako bi bili efikasniji i efektniji, (b) koristiti semantičke i stilističke sličnosti dokumenata i (c) pronalaziti načine izvlačenja znanja iz njih. Kako bi se smanjila kompleksnost utvrđivanja mjera semantičke sličnosti dokumenata te lakše organizirale informacije i znanje sadržane u njima, tadašnji istraživači često su koristili ontologije – formalne sustave koji opisuju kategorije objekata, pojmove i njihove međusobne odnose (Harispe i sur., 2014; Leroy i Rindflesch, 2005; Patwardhan i sur., 2003), posebno kada je riječ o problemu (engl.) *word-sense disambiguation* (WSD), tj. kada riječ ima više značenja ovisno o kontekstu. Pored ontologija, često se pokušavao koristiti i *WordNet* (Fernando i Stevenson, 2008), leksička baza podataka i jezični resurs koji organizira riječi i njihove značenjske veze u obliku sinonima (sličnih riječi) i hiperonima (nadređenih pojmova) (Princeton University, 2010). Koristi se za poboljšanje razumijevanja značenja riječi, traženje sinonima, otkrivanje semantičkih veza između riječi i za druge jezične zadatke. Tijekom ovog istraživanja višestruko se pokušavalo iskoristiti WordNet za poboljšanje rezultata utvrđivanja sličnosti tekstova, no bez vidljivih uspjeha.

Corley i Mihalcea koristili su metodu temeljenu na znanju koja spaja metriku semantičke sličnosti riječi u metriku semantičke sličnosti teksta (Corley i Mihalcea, 2005). Koristili su šest mjernih podataka sličnosti riječi: Leacock i Chodorow, Lesk, Wu i Palmer, Resnik, Lin te Jiang i Conrath (temeljene na WordNetu), a implementacija tih metrika definirana je između pojmova, a ne riječi. Postigli su točnost od 68.8% uz F-mjeru od 77.7%.

Nadalje, Mihalcea i sur. (2006) predstavili su kombiniranu metodu za mjerenje semantičke sličnosti tekstova koja, prema autorima, nadmašuje pristup temeljen na vektorima.

1 *Shingling* dijeli tekst na manje dijelove, obično nizove od nekoliko riječi, kako bi se usporedila sličnost između tekstova. *Fingerprinting* izdvaja ključne značajke teksta, poput najčešćih riječi ili kombinacija riječi, kako bi se stvorio jedinstveni sažetak za brzu usporedbu s drugim tekstovima.

Njihova kombinirana metoda za mjerenje sličnosti riječi koristi dvije metode temeljene na korpusu i šest metoda temeljenih na znanju. Kombiniranu metodu koristili su „za izvođenje metrike sličnosti tekst-tekst" (Mihalcea i sur., 2006). Mihalcea je također sudjelovala u istraživanju Corley i sur. (2007), gdje su istraživači koristili „šest različitih metrika, odabranih uglavnom zbog njihove opažene učinkovitosti u primjenama obrade prirodnog jezika [...] i zbog njihove relativno visoke računalne učinkovitosti" (Corley i sur., 2007). To su metrike sličnosti riječi temeljene na *WordNetu*, kojima su se bavili u svojim radovima primjerice Leacock i Chodorow, Lesk, Wu i Palmer, Resnik, Lin i Jiang i Conrath, više puta spominjani i citirani u ovome radu.

Fernando i Stevenson (2008) koristili su algoritam za identifikaciju parafraziranja koji intenzivno koristi informacije o sličnosti riječi izvedene iz *WordNeta*. Za računanje sličnosti između parova rečenica koristili su kosinusnu sličnost između vektora koji predstavljaju rečenice te semantičku matricu sličnosti koja sadrži informacije o sličnosti parova riječi izvedene iz šest metrika temeljenih na hijerarhiji *WordNeta* (Fernando i Stevenson, 2008). Rečenice su predstavljali kao binarne vektore (s elementima jednakima 1 ako je riječ prisutna u rečenici te 0 ako nije prisutna).

Callison-Burch (2008) je koristio dvojezične parove izraza[1] i izmijenio uobičajeni algoritam za ekstrakciju izraza kako bi izvukao oznake[2] izraza iz parova izraza, omogućujući time generiranje parafraziranog teksta ili njegovo otkrivanje u dvojezičnim korpusima (Callison-Burch, 2008). Postavio je i sintaktička ograničenja na uobičajenu tehniku parafraziranja koja ekstrahira parafraze te zahtjev da izrazi budu istog sintaktičkog tipa kao i izrazi koje parafraziraju.

Chong i sur. (2010) primijenili su nekoliko tehnika NLP na kratke odlomke kako bi analizirali strukturu teksta i automatski identificirali plagirane tekstove. Dokazali su da NLP tehnike mogu poboljšati točnost otkrivanja plagijata, iako imaju poteškoće kod otkrivanja prikrivenih plagijata (M. Chong i sur., 2010).

Socher i sur. (2011) kreirali su algoritam za nenadzirano učenje značajki temeljen na nenadziranom rekurzivnom *autoenkoderu* koji uči vektore značajki za izraze u sintaktičkim stablima. Te se značajke koriste za mjerenje sličnosti između riječi i izraza u dvjema

1 Izraz, sintagma, fraza.

2 Oznake izraza odnose se na dodatne informacije koje se pridodaju izrazu kako bi se opisala njegova funkcija, značenje ili drugi aspekti potrebni za određeni zadatak, poput označavanja sintaktičke kategorije ili semantičke uloge.

rečenicama (Socher i sur., 2011).

Šarić i sur. (2012) koristili su pristupe temeljene na znanju i pristupe temeljene na korpusu. Sve njihove mjere sličnosti riječi temeljene na znanju zasnivaju se na *WordNetu*. Kako bi izračunali ocjenu sličnosti za par riječi, autori su uzimali maksimalnu ocjenu sličnosti za sve moguće parove izraza – *WordNet* sinonimnih skupova. Autori su koristili najniži zajednički nadređeni pojam (engl. *Lowest Common Subsumer*, LCS) dvaju izraza koji predstavlja najniži čvor u hijerarhiji *WordNeta*, a koji je hiponim za oba izraza. Za izračunavanje mjera sličnosti *PathLen* i *Lin* koristili su programsku biblioteku *nltk*. Za sličnost riječi temeljenu na korpusu autori su koristili distribucijske modele leksičke semantike, gdje „izvođenje semantičke sličnosti između dvije riječi odgovara usporedbi tih distribucija" (Šarić i sur., 2012). Naime koristili su latentnu semantičku analizu (LSA) nad velikim korpusom za procjenu distribucija.

Chong (2013) je proveo opsežno istraživanje isprobavajući sve metode i pristupe za mjerenje sličnosti tekstova u malim i velikim korpusima. Autor je u svojem empirijskom istraživanju došao do zaključka da je strojno učenje ključno za bilo kakav postupak automatskog otkrivanja plagijata (M. Y. M. Chong, 2013).

Iako su ideje i istraživanja korištenja dubokih neuronskih mreža za obradu prirodnog jezika postojale i ranije, njihov je broj te rezultatski domet znatno povećan od 2013. godine pojavom *Word2Veca*, prvoga jezičnog modela temeljenog na dubokom učenju (Mikolov i sur., 2013a), mada su se obrisi novog pristupa dubokom učenju (preciznije njegova revitalizacija zbog otkrića novih algoritama i principa te zbog povećanja računalne moći) vidjeli i godinu prije u radovima istog autora (Mikolov, 2012; Mikolov i Zweig, 2012). Prethodno, većina istraživanja na modeliranju rečenica usredotočila se na značajke poput preklapanja n-grama (Clough i Stevenson, 2011), sintaktičkih značajki (Corley i sur., 2007; Pennington i sur., 2014) i značajki temeljenih na strojnom prevođenju (Clough i Stevenson, 2011). Metode dubokog učenja usmjerile su pozornost istraživača prema tzv. distribuiranim reprezentacijama teksta, poznatijima kao vektorske reprezentacije koje omogućuju modelima prepoznavanje i korištenje semantičkih odnosa između riječi i fraza. Ta je promjena dovela do značajnog napretka u razumijevanju i obradi prirodnog jezika jer je fokus pomaknut s površinskih značajki teksta na dublje semantičke strukture. Za sličnost riječi i rečenica predložene su razne arhitekture temeljene na dubokim neuronskim mrežama[1]. Mikolov i sur. (2013b)

1 Strategija je to na koju oslanja i istraživanje opisano u ovom radu.

predstavili su općenito primjenjivu metodu pomaka vektora za identificiranje jezičnih pravilnosti u kontinuiranim prostornim prikazima riječi (Mikolov i sur., 2013b). Pokazali su da su prikazi riječi naučeni modeliranjem jezika pomoću rekurentnih neuronskih mreža (engl. *recurrent neural networks*) (Mikolov i sur., 2011) vrlo dobri alati za predstavljanje tih pravilnosti. Le i Mikolov (2014) izumili su „nenadzirani algoritam učenja koji uči vektorske prikaze za tekstove promjenjive dužine, poput rečenica i dokumenata. Vektorski prikazi se uče kako bi predvidjeli okolne riječi u kontekstima uzorkovanima iz odlomka" (Le i Mikolov, 2014).

Duboko učenje (engl. *Deep Learning*, DL) koristili su Banea i sur. (2014) eksperimentirajući „s tradicionalnim mjerama temeljenim na znanju, kao i s novim mjerama temeljenim na korpusu koje se baziraju na paradigmi dubokog učenja, u kombinaciji s različitim stupnjevima proširenja konteksta" (Banea i sur., 2014).

Socher (2014) je uveo rekurzivne metode dubokog učenja koje su varijacije i proširenja nenadziranih i nadziranih rekurzivnih neuronskih mreža. Ta metoda koristi ideju hijerarhijske strukture teksta i kodira dva vektora riječi u jedan vektor pomoću autoenkoderskih mreža. Socher je također predstavio mnoge varijacije dubokih funkcija, kao što su kombinacije rekurzivne neuronske mreže i matrično-vektorske rekurzivne neuronske mreže (Socher, 2014).

Kong i sur. (2014) pokušali su otkrivati „jako prikriveno" plagiranje koristeći model logičke regresije. Predloženi model integrirao je leksičke, sintaktičke, semantičke i strukturne značajke koje su izvučene iz sumnjivih dokumenata i izvornih dokumenata (Kong i sur., 2014).

Gipp (2014) je primijenio metodu otkrivanja plagiranja temeljenu na citatima koja ne uzima u obzir samo sličnost tekstova već koristi obrasce citiranja unutar dokumenata kako bi identificirao njihovu potencijalno sumnjivu sličnost (Gipp, 2014).

Kim i sur. (2014) opisali su niz eksperimenata s konvolucijskim neuronskim mrežama izgrađenima na temeljima *Word2Vec* modela i zaključili da su nenadzirano predtrenirani vektori riječi vrlo korisni u NLP zadacima (Kim, 2014).

Yin i Schuetze (2015) predložili su novu arhitekturu dubokog učenja Bi-CNN-MI[1] za

1 „Bi-CNN" označava dvostruke konvolucijske neuronske mreže korištene u siamskom okviru, dok „MI" označava multigranularne značajke interakcije. Siamski okvir se odnosi na arhitekturu neuronske mreže koja se sastoji od dvije identične podmreže koje dijele iste parametre i težine. Ove dvije podmreže obrađuju dva različita ulaza i njihov zadatak je usporediti te ulaze kako bi se odredila njihova sličnost.

identifikaciju parafraziranja uspoređivanjem rečenica na višestrukim razinama granularnosti, koristeći konvolucijsku neuronsku mrežu (CNN) i modelirajući interakcijske značajke na svakoj razini (Yin i Schuetze, 2015). Te su značajke zatim ulaz za logistički klasifikator za identifikaciju parafraziranja.

Gharavi i sur. (2016) predložili su metodu temeljenu na dubokom učenju za otkrivanje plagijata na perzijskom jeziku. Riječi su predstavljene kao višedimenzionalni vektori, a jednostavne metode agregacije koriste se za kombiniranje vektora riječi za reprezentaciju rečenica. Usporedbom reprezentacija izvornih i sumnjivih rečenica parovi rečenica s najvećom sličnosti i s graničnom vrijednošću kosinusne sličnosti od 0.3 smatraju se kandidatima za plagijat. Odluka o tome je li riječ o plagijatu donosi se pomoću druge razine metode evaluacije, korištenjem Jaccardove mjere sličnosti s graničnom vrijednošću od 0.2 (Gharavi i sur., 2016).

Thompson i Bowerman (2017) su za otkrivanje najčešćih tehnika korištenih u parafraziranju tekstova (leksička zamjena, umetanje/brisanje, preuređivanje riječi i izraza) predložili metode temeljene na korpusu i kombinirali ih u model za otkrivanje parafraziranja (Thompson i Bowerman, 2017).

Agarwal i sur. (2017) razvili su robusni model za otkrivanje parafraziranja temeljen na tehnikama dubokog učenja. Predložili su hibridnu duboku neuronsku arhitekturu sastavljenu od konvolucijske neuronske mreže CNN i modela dugoročnoga kratkoročnog pamćenja LSTM (engl. *long short-term memory*) dodatno poboljšanu modulom za sličnost parova riječi (Agarwal i sur., 2017).

J. Zhou i sur. (2018) predložili su model dubokog učenja za identifikaciju parafraziranja temeljen na značajkama sličnosti parova jedinica izvađenih iz zadanih parova teksta putem modela konvolucijske neuronske mreže CNN i značajki semantičke kontekstne korelacije temeljenih na CNN-u i LSTM-u (J. Zhou i sur., 2018).

Z. Li i sur. (2018) izradili su generator parafraziranih tekstova i evaluator za njih. Za treniranje evaluatora koristili su obrnut pristup u odnosu na onaj korišten za treniranje generatora, zbog čega postoje određene rezerve prema toj evaluaciji. (Z. Li i sur., 2018).

W. Wu i sur. (2018) primijetili su da potrošnja memorije prethodnih modela za enkodiranje rečenica raste kvadratno s dužinom rečenice, a sintaktičke informacije su zanemarene. Stoga su napravili model koji može iskoristiti sintaktičke informacije za univerzalno enkodiranje rečenica, filtrirati udaljene i nepovezane riječi te se usredotočiti na

modeliranje interakcije između semantički i sintaktički važnih riječi (W. Wu i sur., 2018).

Zablocki i sur. (2018) predložili su multimodalni (tekst i slika) kontekstni pristup za učenje vektorskih reprezentacija riječi. U svojim eksperimentima pokazali su da su vizualna okruženja objekata i njihove relativne lokalizacije vrlo informativni za izgradnju reprezentacija riječi (Zablocki i sur., 2018).

Ramaprabha i sur. (2018) istraživali su kako se konvolucijske neuronske mreže i rekurentne neuronske mreže, poput LSTM modela, mogu koristiti za kodiranje i uspoređivanje rečenica s ciljem određivanja njihove semantičke sličnosti. Smatraju da su otkrivanje parafraziranja te razumijevanje teksta i pitanja izazovni zadaci te kako duboko učenje može pomoći u njihovu rješavanju (Ramaprabha i sur., 2018).

Devlin i sur. (2018) iz *Googlea* predstavili su novu tehniku nazvanu *masked LM* (MLM) koja omogućava dvosmjerno treniranje jezičnih modela. Koristeći tu tehniku, razvili su BERT model koji se koristi u širokom spektru NLP zadataka. (Devlin i sur., 2018).

Yang i sur. (2019) primijetili su da „BERT zanemaruje ovisnost između maskiranih pozicija i pati od nesklada između predtreniranja i finog podešavanja". Da bi to prevladali, predložili su *XLNet*, generaliziranu autoregresivnu metodu predtreniranja koja (1) omogućava učenje dvosmjernih konteksta maksimiziranjem očekivane vjerojatnosti preko svih permutacija redoslijeda faktorizacije i (2) prevladava ograničenja BERT-a zahvaljujući svojoj autoregresivnoj formulaciji (Yang i sur., 2019).

Kako bi poboljšali otkrivanje parafraziranja, El Desouki i sur. (2019) predložili su model koji kombinira pristup sličnosti tekstova s pristupom dubokog učenja koristeći model *skip-thought* koji se temelji na dubokom učenju koji koriste za dobivanje semantičkog vektora svake rečenice (El Desouki i sur., 2019), a zatim mjere vektorsku sličnost između dobivenih semantičkih vektora koristeći više mjera sličnosti zasebno i u kombinaciji, na način kako su to prethodno predstavili Gomaa i Fahmy (2017) koji su različite algoritme sličnosti spojili unutar *SimAll* alata (Gomaa i Fahmy, 2017).

Ahmed i sur. (2019) predložili su poboljšanje modela Tree-LSTM uvođenjem mehanizma pažnje. Autori tvrde da tradicionalni Tree-LSTM modeli jednako tretiraju sve riječi unutar podstabla, zanemarujući potencijalne razlike u važnosti. Stoga su uveli mehanizam pažnje kako bi se odredila važnost svake komponente podstabla prilikom izgradnje cijelog stabla, bilo semantički ili sintaktički, a autori primjenjuju model pažnje na zadatak semantičke sličnosti, gdje model treba dati ocjenu sličnosti između dviju rečenica.

Njihovi eksperimenti pokazali su da predloženi modeli pažnje nadmašuju tradicionalne modele Tree-LSTM u zadatku otkrivanja semantičke sličnosti. (Ahmed i sur., 2019).

Tenney i sur. (2019) zaključili su da su postojeći modeli trenirani na jezičnom modeliranju vrlo korisni za sintaktičke zadatke, ali nude mala poboljšanja na semantičkim zadacima (Tenney i sur., 2019).

Reimers & Gurevych (2019) koristili su Sentence-BERT, varijantu modela BERT prilagođenu za generiranje vektorskih reprezentacija rečenica. Korištenjem sijamskih BERT mreža, njihov model može mjeriti semantičku sličnost između rečenica. Autori konstatiraju da su dobiveni rezultati takvih kalkulacija lošiji od rezultata dobivenih korištenjem vektorskih reprezentacija modela GloVe (Reimers i Gurevych, 2019).

Shuang i sur. (2020) usmjerili su se na rješavanje problema višeznačnosti kao prepreke za uspješnije otkrivanje parafraziranja. Naime, u modelima dubokog učenja jedna riječ uglavnom ima jedinstvenu vektorsku reprezentaciju, bez obzira na kontekst. Stoga su predložili konvolucijsko-dekonvolucijsku vektorsku reprezentaciju riječi CDWE, višestruki prototip fuzijskog stvaranja vektorskih reprezentacija riječi koji spaja kontekstno specifične informacije i specifične informacije potrebne za rješavanje zadatka (Shuang i sur., 2020).

El Mostafa i Benabbou (2020) proveli su komparativno istraživanje otkrivanja plagijata. Zaključili su da većina istraživanja koristi granularnost riječi i metodu Word2Vec za prikaz vektora riječi. To je, prema autorima, slaba točka tih istraživanja jer ne odražavaju istinska značenja rečenica (El Mostafa i Benabbou, 2020).

Chandrasekaran i Mago (2022) načinili su pregledni rad, istraživanje „najizazovnijeg zadatka" u području obrade NLP-a: mjerenja semantičke sličnosti između dva (isječka) teksta. Autori razlikuju metode temeljene na znanju, korpusu, dubokim neuronskim mrežama i hibridne metode. Metode temeljene na znanju uzimaju u obzir stvarno značenje teksta, ali nisu prilagodljive za različite domene i jezike. Metode temeljene na korpusu imaju statističku pozadinu i mogu se implementirati na različitim jezicima, ali ne uzimaju u obzir stvarno značenje teksta. Metode temeljene na dubokim neuronskim mrežama pokazuju bolje performanse, ali zahtijevaju velike računalne resurse i nedostaje im interpretabilnost. Hibridne metode nastaju kako bi se iskoristile prednosti i kompenzirali nedostaci različitih metoda. Autori zaključuju da, iako postoje obećavajući rezultati, postoje jasne potrebe za izgradnju semantički svjesnijih vektorskih reprezentacija riječi, za određivanje ravnoteže između računalne učinkovitosti i performansi te za idealni korpus (Chandrasekaran i Mago, 2022).

Slično pregledno istraživanje objavili su Han i sur. (2021), uz dvije različitosti. Prva je ta što potonji ne razmatraju hibridni pristup kao perspektivan, a druga je različitost u tome što potonje istraživanje ima i eksperimentalne rezultate koji pokazuju da su mjere semantičke sličnosti temeljene na DL-u dale bolje rezultate od tradicionalnih metoda, pri čemu posebno model BERT postiže najbolje performanse u mjerama sličnosti kratkog teksta i s performansama koje daleko premašuju druge modele (Han i sur., 2021).

Amur i sur. (2023) ističu potrebu za razvojem naprednijih dubokih modela učenja koji mogu bolje uhvatiti kontekstne informacije u kratkim tekstovima. Također preporučuju stvaranje specijaliziranih podatkovnih skupova koji su prilagođeni isključivo za analizu kratkog teksta (Amur i sur., 2023).

Bali i sur. (2024) istražuju područje otkrivanja i analize semantičke sličnosti tekstnih dokumenata te daju pregled različitih metoda za otkrivanje semantičke sličnosti u tekstnim dokumentima, uključujući kosinusnu sličnost, Jaccardovu sličnost, latentnu semantičku analizu (LSA) i BERT. Autori istražuju kako te metode mogu izmjeriti sličnost između dokumenata na temelju njihova značenja, a ne samo doslovnog podudaranja riječi, no izvan pregleda područja ne nude novitet proizašao iz njihova istraživanja (Bali i sur., 2024).

Postojeća istraživanja nisu se dovoljno približila rješenju problema otkrivanja **semantičke** sličnosti tekstova. Stoga postoji velika potreba za novim pristupima otkrivanju prikrivenih plagijata, posebno parafraziranja, i to takvima koji se temelje na dubokom učenju koje koriste i svi postojeći veliki jezični modeli. S obzirom na inherentnu semantiku vektorskog prostora nastalog treniranjem neuronskih mreža velikom količinom često probranih i kvalitetnih sadržaja – u ovom slučaju teksta, specifičnih za neku domenu ili čak opće prirode, trebao bi postojati način na koji se mogu pouzdano otkrivati semantičke sličnosti tekstova na dostatnoj razini uspješnosti kako bi se mogli otkrivati ne samo jednostavni plagijati već i oni prikriveni. Ovo istraživanje i njegovi rezultati, prikazani u nastavku, to i potvrđuju.

2.6.2. Istraživanja semantičke sličnosti tekstova

Do sada neriješeni problemi u domeni otkrivanja plagijata su prepoznavanje prikrivenih plagijata nastalih parafraziranjem, prevođenjem i plagiranjem ideja. Posebno je teško utvrđivanje prikrivenih plagiranja na automatizirani način – pomoću programskih sustava. I premda neki suvremeni programski sustavi poput Turnitina uključuju module za

prevođenje teksta, uspješnost detekcije plagiranja prijevodom je i dalje nedostatna. Automatsko otkrivanje duplikata i plagijata izvan očitog preklapanja niza pojmova, riječi i/ili rečenica, zahtijeva prepoznavanje semantičke sličnosti (Marsi i Krahmer, 2013).

S druge strane, ukoliko je iz dvaju dokumenata moguće prepoznati istu semantičku informaciju, onda ih možemo smatrati semantički sličnima te uz preduvjet nedostatka referenciranja, ustvrditi slučaj plagiranja (Al-Shamery i Gheni, 2016). Štoviše, da bi na automatizirani način bilo uopće moguće utvrditi prikriveno plagiranje, dakle ono koje nije očito preklapanje istih pojmova, nužno je prepoznati semantičku sličnost (Marsi i Krahmer, 2010), a da bi to bilo moguće, sustav za mjerenje semantičke sličnosti mora imati informaciju o riječima i njihovu značenju. No čak i tada, samo utvrđivanje semantičke sličnosti nije dostatno – potrebno je utvrditi kolika je mjera te sličnosti.

Kao što dokazuju Mihaelcea i suradnici, metode prepoznavanja semantičke sličnosti trebale bi biti učinkovitije od metoda koje se temelje na jednostavnom leksičkom podudaranju te koristiti model vektorskog prostora (Mihalcea i sur., 2006). Te starije, uobičajene ili klasične metode otkrivanja plagiranja uglavnom se temelje na uspoređivanju malih jedinica teksta, poput n-grama (n slova, n-riječi), rečenica ili odlomaka, dok suvremene metode traže (često i heurističke) putove otkrivanja i uspoređivanja semantičke sličnosti.

Banea i suradnici (2014) smatraju da je otkrivanje semantičke sličnosti tekstova jedna je od ključnih komponenti primjene NLP-a u domenama izvlačenja informacija (engl. *Information Retrieval*), relevantnosti povratne informacije (engl. *Relevance Feedback*), klasifikacije teksta (engl. *Text Classification*), razlučivanja smisla riječi (engl. *Word Sense Disambiguation*), sažimanja (engl. *Summarization*), automatske evaluacije strojnog prijevoda (engl. *Automatic Evaluation of Machine Translation*) i kod detekcije plagiranja (Banea i sur., 2014).

Metode utvrđivanja semantičke sličnosti mogu djelovati na različitim razinama granulacije teksta, tj. semantička se sličnost može utvrditi između tekstova različitih veličina (Rus i sur., 2013): sličnosti simetričnih veličina riječ-za-riječ, izraz-za-izraz, rečenica-za-rečenica, odlomak-za-odlomak i sličnost dokument-za-dokument. Također se može utvrđivati i sličnost nesimetričnih granularnosti teksta, poput sličnosti rečenica-za-odlomak, no prilikom utvrđivanja plagiranja najčešće se uspoređivanje vrši na nivou rečenice (Czerski i sur., 2015). Za te je dakle, tipične metode, prema Mihalcei i suradnicima, uobičajeni pristup pronalaženja sličnosti između dva tekstna segmenta jednostavna leksička podudarnost koja stvara ocjenu

sličnosti temeljem broja leksičkih jedinica koje se pojavljuju u oba ulazna segmenta (Mihalcea i sur., 2006). Unaprjeđenja te jednostavne metode su korjenovanje (engl. *stemming*), uklanjanje zaustavnih riječi (engl. *stop-words*), označavanje dijelova teksta, traženje najdužih podudarnih segmenata te korištenje raznih faktora težine i faktora normalizacije.

Metode utvrđivanja semantičke sličnosti dvaju tekstova koriste (Banea i sur., 2014; Mihalcea i sur., 2006) dvije klase mjera sličnosti: temeljem korpusa i temeljem znanja, a obje klase tradicionalno koriste model vektorskog prostora (engl. *Vector Space Model*). Naime, da bi se računalom moglo učinkovito obrađivati tekst, on se mora pretvoriti u neki numerički zapis.

Semantičke metode utvrđivanja sličnosti tekstova imaju široku primjenu u različitim područjima (Marsi i Krahmer, 2010). Koriste se za automatsko sažimanje tekstova, prepoznavanje semantičke sličnosti radi automatske detekcije duplikata i plagijata, kao i u automatskim sustavima upita i odgovora gdje se grupiraju semantički slični odgovori. Nadalje, primjenjuju se za spajanje dokumenata s istim, ali revidiranim tekstom, te za prepoznavanje tekstnih implikacija. Marsi i Krahmer ističu da semantičke metode utvrđivanja sličnosti tekstova koriste dva glavna pristupa (Marsi i Krahmer, 2010). Prvi i osnovni temelji se na mjerama sličnosti nizova znakova, poput Levenshteinove udaljenosti ili Jaccardovog koeficijenta sličnosti. Drugi pristup uključuje duboku semantičku analizu teksta i primjenu metodologije formalnog zaključivanja. Prvi je pristup brz, ali se ne može nositi sa sofisticiranim slučajevima plagiranja. Drugi je pristup obično suviše zahtjevan u pogledu potrebnih računalnih resursa ili utroška vremena. Pored toga, (a) nije dovoljno robustan jer ovisi o razvijenosti obrade jezika te (b) ne garantira pronalaženje svih semantičkih i logičkih veza između uspoređivanih dokumenata. Marsi i Krahmer predlažu analizu semantičke sličnosti između rečenica poravnavanjem njihovih sintaktičkih stabala, pri čemu se svaki čvor usklađuje s najsličnijim čvorom u drugom stablu, ako takav postoji (Marsi i Krahmer, 2010).

Clough (2000) definirao je niz kriterija za utvrđivanje semantičke sličnosti tekstova koji obuhvaćaju razlike u rječniku, promjene rječnika unutar istog teksta, nesuvislost teksta, identičnost interpunkcije, količinu sličnosti između tekstova, iste pravopisne greške, jednaku statističku distribuciju riječi, istu sintaksu, istu dužinu rečenica, isti slijed tema, konzistentno korištenje istih fraza i izraza, frekvenciju riječi, preferencije korištenja kratkih ili dugih rečenica, čitkost teksta te referencije koje nedostaju u popisu literature (Clough, 2000).

Određivanje semantičke sličnosti tekstova (Harispe i sur., 2014; Marsi i Krahmer, 2010; Zervanou i sur., 2014) zadire u područje računalne analize prirodnog jezika pa se u tom kontekstu primjenjuju metode umjetne inteligencije, računalne obrade prirodnih jezika (NLP), dubinske analize podataka (teksta), metode stilometrijske analize teksta (Brennan i Greenstadt, 2009; Zu Eissen i Stein, 2006), metode izvlačenja i prezentacije znanja i značenja iz dokumenata, podataka i prirodnih jezika (Jakupović i sur., 2013; Koch i sur., 2014; Pavlić, Jakupović, i sur., 2013; Pavlić, Meštrović, i sur., 2013; Rajagopal i sur., 2013), grafičke metode prezentacije znanja poput BG (engl. *Basic Conceptual Graphs*) i NOK (engl. *Nodes of Knowledge*), podatkovni modeli, semantičke mreže, neuronske mreže, metoda MultiNets, HSF metoda za predstavljanje uzoraka u prirodnim jezicima.

Mihalcea i sur. (Mihalcea i sur., 2006) kažu da „postoji velik broj mjera semantičke sličnosti riječi koje koriste pristupe koji su ili temeljeni na znanju (Leacock i sur., 1998; Z. Wu i Palmer, 1994) ili na korpusu (P. Turney, 2001). Takve mjere uspješno su primijenjene na zadatke obrade jezika poput otkrivanja malapropizama (Budanitsky i Hirst, 2001), razjašnjavanja značenja riječi (Patwardhan i sur., 2003) i identifikacije sinonima (P. Turney, 2001). Za semantičku sličnost temeljenu na tekstu, možda su najčešće korišteni pristupi aproksimacije dobivene kroz proširenje upita kako se izvodi u pretraživanju informacija (Voorhees, 1993), ili metoda latentne semantičke analize (Landauer i sur., 1998) koja mjeri sličnost tekstova iskorištavajući automatski dobivene odnose između riječi drugog reda iz velikih zbirki tekstova".

Metode otkrivanja semantičke sličnosti nastoje riješiti trostruki problem otkrivanja (a) leksičkih promjena, (b) promjena strukture teksta i (c) parafraziranja, pri čemu je posljednji ujedno i najsloženiji (M. Y. M. Chong, 2013; Ram i sur., 2014).

2.6.3. Istraživanja mjera sličnosti/udaljenosti

Istraživači su razvijali ili primjenjivali različite mjere sličnosti teksta kako bi kvantificirali stupanj sličnosti između dokumenata. Neke od najranijih mjera uključivale su Jaccardov[1] koeficijent (Jaccard, 1912), Levenshteinovu udaljenost (Levenshtein, 1966) i kosinusnu sličnost (Salton i sur., 1975) koje su se pokazale korisnima u različitim primjenama, uključujući detekciju plagijata. Levenshteinova udaljenost mjeri minimalni broj

1 Izvorno korišten u botanici, Jaccardov indeks je pronašao široku primjenu u različitim područjima, uključujući računalnu lingvistiku i NLP, pa se tako može koristiti za identificiranje potencijalnih slučajeva plagijata.

operacija potrebnih za transformaciju jednog niza znakova u drugi. Jaccardov se koeficijent koristi za usporedbu sličnosti i raznovrsnosti skupova, dok kosinusna sličnost mjeri kut između dva vektora u višedimenzionalnom prostoru, često korištenog za analizu sličnosti tekstova (Perone, 2013).

Harispe i sur. (2013, 2017), a potom i Luu i sur. (2020) dali su opsežan pregled mjera udaljenosti i sličnosti teksta s ciljem pronalaženja sličnosti *web*-stranica, ali bez korištenja vektorskog prikaza teksta, osim za mjeru kosinusne sličnosti gdje je to nužno. Zaključili su da kosinusna sličnost nije najbolji izbor u nekim slučajevima jer je manje učinkovita (Harispe i sur., 2013, 2017; Luu i sur., 2020).

Sidorov i sur. (2014) redložili su mjeru sličnosti koju su nazvali meka kosinusna sličnost (engl. *Soft Cosine*), koja uz osnovnu kosinusnu sličnost uzima u obzir sličnost između značajki koje su poznate i ne trebaju se učiti iz podataka. Kada nema sličnosti između značajki, tada je meka kosinusna sličnost jednaka standardnoj kosinusnoj sličnosti (Sidorov i sur., 2014).

Charlet i Damnati (2017) koristili su model Word2Vec u kombinaciji s mekom kosinusnom sličnošću kako bi dobili mjeru semantičke sličnosti tekstova. Istraživali su nenadzirane mjere sličnosti koristeći model Word2Vec treniran na engleskoj Wikipediji, s 300 dimenzija vektorskog prostora. Pokušali su varirati broj dimenzija, ali to nije donijelo značajnu razliku (Charlet & Damnati, 2017).

Mohammad i Hirst (2012a, 2012b) pokušali su dobiti neke mjere semantičke sličnosti, pa su se fokusirali na korištenje distribucijskih mjera i mjera temeljenih na ontologiji. Pri tome su se suočili sa značajnim istraživačkim problemima (Mohammad & Hirst, 2012a, 2012b).

Vrbanec i Meštrović (2017, 2020, 2021b) nastojeći izvući semantičke sličnosti iz tekstova, temeljene na korpusu, pokušali su identificirati sredstva, alate i mjere za to, a zatim su proveli eksperimente u kojima su računali sličnosti tekstova koristeći nekoliko modela DL, temeljenih na korpusu, uparenih s mjerom kosinusne sličnosti. Dobili su obećavajuće rezultate koji potiču na daljnja istraživanja metoda temeljenih na jezičnim modelima dubokog učenja. (Vrbanec & Meštrović, 2017, 2020, 2021b).

2.6.4. Nedostaci postojećih istraživanja

Tradicionalne mjere sličnosti, poput Jaccardova koeficijenta i Levenshteinove

udaljenosti, pokazuju ograničenja u otkrivanju semantičke sličnosti tekstova i prikrivenih plagijata. Jaccardov koeficijent ne prepoznaje semantičke sličnosti jer se fokusira na površinsku prisutnost riječi. Levenshteinova udaljenost također mjeri samo površinske promjene u tekstu, ne uzima u obzir semantičke odnose i osjetljiva je na dužinu teksta. *Hierarchical Dirichlet Process* složen je i teško interpretativan model, fokusiran na otkrivanje tema, a ne sličnosti između dokumenata. *Greedy String Tiling* pronalazi doslovne podudarnosti podnizova znakova, ignorirajući kontekstualne i semantičke promjene. *Greedy Word Tiling* također pronalazi doslovne podudarnosti riječi, pa ne može utvrditi sličnost tekstova u slučaju korištenja sinonima ili kod parafraziranja. Obje *Greedy* metode osjetljive su na zamjene riječi ili promjene redoslijeda rečenica. *Latent Dirichlet Allocation* detektira teme unutar korpusa, a ne ukazuje na semantičku sličnost između tekstova. *LogEntropy* je metoda za ponderiranje riječi koja naglašava rijetke, ali informativne riječi, no ne uzima u obzir kontekstne ili semantičke odnose između riječi, a dodatne su joj slabe točke sinonimi i parafrazirani tekst. *Random Projections* / *Random Indexing* metoda je koja smanjuje dimenzionalnost podataka, ali pri tome gubi semantičke informacije. *Term Frequency – Inverse Document Frequency* (TF-IDF) dobro rangira ključne riječi, ali je neosjetljiva na redoslijed riječi i ne može detektirati ni sinonime ni semantičke veze, što otežava detekciju semantički sličnih, ali drugačije formuliranih tekstova. Svih navedenih devet „klasičnih" metoda pokazuju značajna ograničenja u otkrivanju složenih oblika plagijata, posebno kada je riječ o prikrivenim plagijatima i semantičkim sličnostima. Tražeći površinske podudarnosti zanemaruju dublje semantičke veze i kontekstne razlike između tekstova. Stoga imaju ograničenu sposobnost prepoznavanja sofisticiranih parafraziranih tekstova i sinonima koje su važne za otkrivanje prikrivenog plagiranja. Za učinkovitije otkrivanje prikrivenih plagijata bilo je potrebno razviti naprednije metode koje bolje prepoznaju semantičke odnose i kontekstne informacije.

Istraživanja koja su koristila jezične modele poput Word2Vec (Mikolov i sur., 2013a), Doc2Vec (Le i Mikolov, 2014), FastText (Bojanowski i sur., 2016) i GloVe (Pennington i sur., 2014), pokazala su nedostatke u prepoznavanju semantičkih sličnosti i otkrivanju prikrivenih plagijata. Ti modeli, iako su predstavljali velik napredak u mogućnosti zadržavanja semantike teksta, ipak nisu dovoljno sofisticirani za prepoznavanje suptilnih razlika i kontekstnih nijansi u tekstovima, što je ključno za preciznu detekciju plagijata. Dakle, tradicionalni modeli,

metode temeljene na korpusu i prvi modeli dubokog učenja[1], nisu pružali zadovoljavajuće rezultate u otkrivanju složenih semantičkih odnosa. Njihova ograničenja u prepoznavanju parafraziranja i sinonima ne omogućuju detekciju prikrivenih plagijata. Veliki jezični modeli poput BERT-a i GPT-a značajno su unaprijedili te mogućnosti iako neki autori navode nedostatke za njihovo korištenje s obzirom na nemogućnost interpretacije rezultata i potrebu za snažnim računalnim resursima. Te dvije primjedbe mogu ograničiti njihovu primjenu u okruženjima s ograničenim resursima, kao i njihovu transparentnost u akademskoj, a možda i u pravnoj domeni. Iako su napredni veliki jezični modeli temeljeni na arhitekturi transformera poboljšali performanse u mnogim NLP zadacima, neispunjena potreba otkrivanja prikrivenih plagijata i sofisticiranog parafraziranja i dalje potiče istraživače na daljnja istraživanja i prilagođavanja tih tehnologija kako bi se postigla njihova maksimalna preciznost.

S obzirom na navedene nedostatke postojećih istraživanja, istraživanje prikazano u ovome radu neophodno je jer se usredotočuje na otkrivanje prikrivenog plagiranja, posebno parafraziranja. Dobivenim rezultatima unaprjeđuje se ova domena novim pristupom, a kombiniranom i novom mjerom semantičke sličnosti pridonosi se razvoju učinkovitijih rješenja omogućujući tako uspješniju detekciju plagijata i parafraziranja u raznim kontekstima.

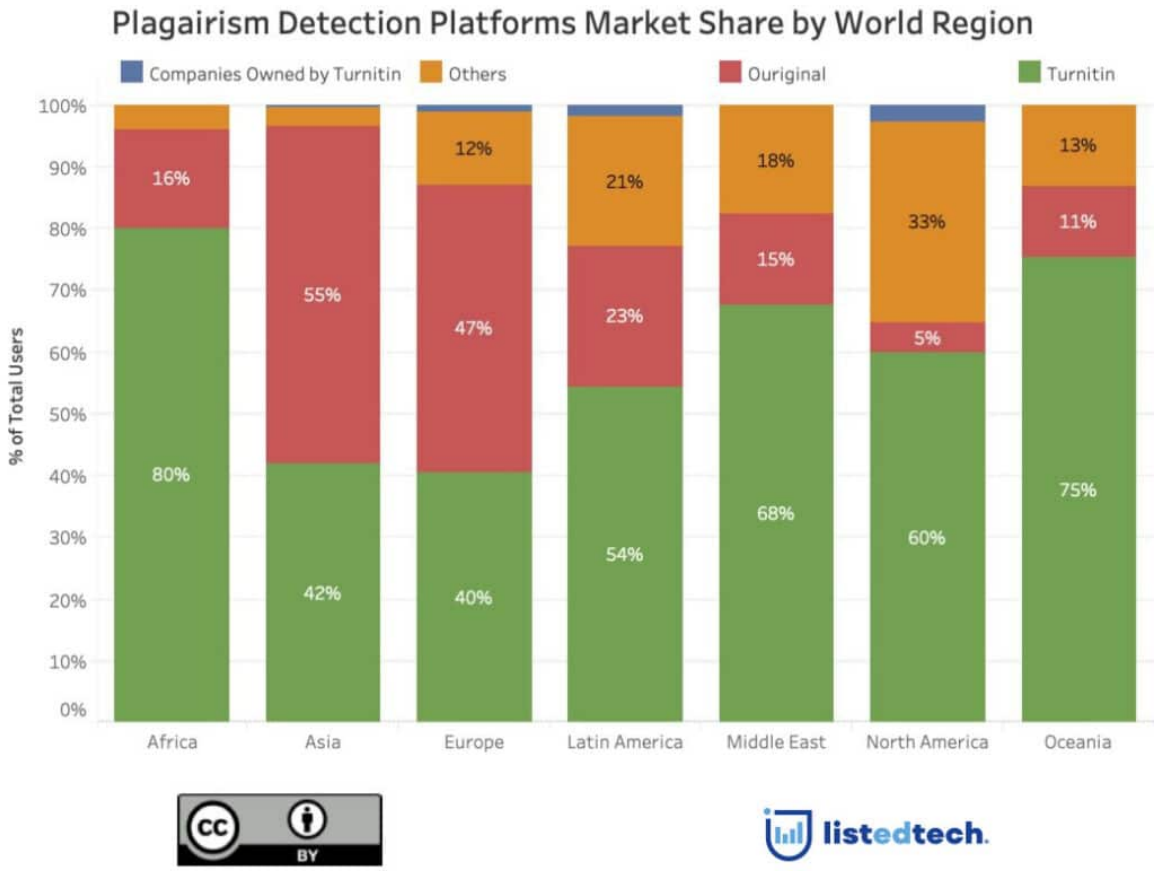
2.7. Razvoj programske podrške za otkrivanje plagiranja

Što se tiče programske podrške za utvrđivanje plagijata, ili kako ih proizvođači programske podrške nazivaju „sustava za provjeru izvornosti”, razvoj računalne tehnologije u 1980-ih i 1990-ih godinama omogućio je stvaranje sofisticiranih alata za otkrivanje plagijata poput *Turnitina*, razvijenog 1997. godine (Turnitin, 2024). Razvila ga je skupina doktoranada sa Sveučilišta California u Berkeleyju, izvorno pod imenom *Plagiarism.org*. *Turnitin* koristi kombinaciju različitih metoda, uključujući usporedbu nizova znakova, analizu učestalosti riječi i analizu sintakse kako bi identificirao potencijalne slučajeve plagijata. *Turnitin* također koristi kombinaciju metoda za pretraživanje internetskih izvora, akademskih baza podataka i vlastitih arhiva radova za identifikaciju potencijalnih plagijata.

Istraživači sa Sveučilišta Stockholm 2000. godine pokrenuli su sustav za otkrivanje plagijata pod nazivom *Urkund*, koji je brzo postao popularan u skandinavskim zemljama

1 *Word2Vec*, *Doc2Vec*, *FastText* i *GloVe*.

zahvaljujući svojoj jednostavnosti uporabe i preciznosti u otkrivanju plagijata. *Urkund* je također bio poznat po svojoj suradnji s obrazovnim institucijama i kontinuiranom razvoju svojih algoritama. *Turnitin* je 2021. godine preuzeo *Urkund* (Turnitin, 2021), preciznije *Ouriginal*, jer su se prije toga 2020. godine spojile dvije tvrtke *Urkund* i *PlagScan* kako bi oblikovale novu tvrtku pod imenom *Ouriginal* (Namrata, 2020). Cilj je bio kombinirati snage obaju sustava kako bi ponudili poboljšane usluge detekcije plagijata i analize teksta. *PlagScan* su pokrenuli njemački istraživači 2009. godine (Bailey, 2011). *Turnitin* je 2012. godine preuzeo i *Ephorus* (Library Learning Space, 2019), čime je učvrstio svoju poziciju kao vodećega globalnog pružatelja usluga za provjeru plagijata. Ukupno gledajući, potezi *Turnitina*, koji je akvizicijama preuzeo druge ključne igrače (neposredne konkurente) kao što su *Urkund* i *PlagScan* (*Ouriginal*) te *Ephorus* doveli su do smanjenja konkurencije i koncentracije moći u rukama jedne tvrtke, tj. do monopola, što prikazuje i sljedeća slika 6 (Ménard, 2021) s istaknutim tržišnim udjelima *Turnitina* i tvrtki u njegovu vlasništvu.



Slika 6. Tržišni udjeli Turnitina u svijetu 2021. godine

Navedeni programski sustavi do sada (ipak) nisu uspijevali pronalaziti semantičku sličnost tekstova, a nije poznato jesu li i koliko njihovi razvijatelji pokušavali u tome. Glavna je prednost postojećih programskih sustava njihov pristup (inače zatvorenim[1]) akademskim bazama znanstvenih i stručnih članaka te dodatno *web*-pretraživanje, no nakon što na navedeni način oblikuju impozantan korpus dokumenata i tekstova s kojima bi trebali usporediti provjeravani dokument, ostaju im klasične metode usporedbe tekstnih nizova, odnosno stringova. Javno objavljeni pokušaji unaprjeđivanja programskih sustava izvan toga okvira uključuju korištenje strojnog prevođenja kako bi što bolje otkrivali plagiranje prijevodom te korištenje *online*-servisa umjetne inteligencije, poput ChatGPT-a, kako bi pokušali utvrditi plagiranje pomoću njihovih rezultata.

3. Računalna obrada prirodnog jezika

U ovome poglavlju opisani su postupci i metode iz područja obrade prirodnoga jezika koji su relevantni za provedeno istraživanje. Na početku su ukratko opisani postupci pretprocesiranja teksta koji se uobičajeno koriste u početnim fazama implementacije metoda i tehnika iz područja NLP-a. Nakon toga dan je pregled pristupa za vektorsku reprezentaciju teksta te pregled velikih jezičnih modela. Na kraju su opisane mjere koje se koriste za određivanje sličnosti tekstova te standardne mjere koje se koriste u postupcima evaluacije metoda iz područja NLP-a.

3.1. Pretprocesiranje teksta

Za potrebe razvoja i/ili primjene bilo kakve metode iz područja NLP-a najčešće je u prvom koraku potrebno provesti pretprocesiranje teksta. Odnosno, tekst koji se ekstrahira iz dokumenata potrebno je prije bilo kakve upotrebe i usporedbe pročistiti i prilagoditi kako bi bio pogodan za daljnju analizu i obradu. Pri tome su na raspolaganju tehnike i metode pretprocesiranja teksta, koje se primjenjuju ovisno o tipu zadataka i metoda koji se potom provode. Pretprocesiranje se može kretati u rasponu od minimalnoga, tj. nekog oblika osnovnog čišćenja teksta (njegova normalizacija) do naprednoga, poput semantičkih

1 Za pristup dijelu značajnih znanstvenih baza članaka potrebno je platiti pretplatu kao pojedinac ili kao ustanova, što ograničava istraživače u siromašnim sredinama i društvima te one koji nisu članovi projektnih skupina s financiranjem iz znanstvenih zaklada, sveučilišta, privrede ili zainteresiranih sponzora/donatora.

transformacija.

Osnovna obrada teksta ili normalizacija može uključivati transformacije iz nestandardnoga kodiranja u utf-8 ili utf-16 kodove, uklanjanje posebnih znakova i interpunkcije koji nemaju semantičko značenje, pretvorbu teksta u mala slova, uklanjanje višestrukih razmaka ili praznih redova, a može uključivati i tokenizaciju tj. razbijanje teksta na manje jedinice. Te su jedinice obično riječi, ali mogu biti i manje od riječi – dijelovi riječi, n-torke[1] znakova, ili veće od riječi – rečenice ili odlomci. U normalizaciju teksta može biti uključena i obrada brojeva: njihovo uklanjanje iz teksta, normalizacija formata ili zamjena tokenom (npr. <NUM>). Nadalje, riječi od jednog znaka mogu se izostaviti iz teksta jer se mogu smatrati neinformativnima.

Leksička i morfološka obrada su postupci obrade teksta s ciljem smanjenja njegove varijabilnosti. Tako je lematizacija svođenje riječi na osnovni (kanonski, leksikografski) oblik, korjenovanje (engl. *stemming*) je svođenje riječi na korijenski morfem uklanjanjem sufiksa. U tu kategoriju postupaka spada i uklanjanje zaustavnih riječi (engl. *stop-words*) tj. eliminacija čestih riječi koje u dostatnoj mjeri ne doprinose značenju teksta, a mogu ometati ili usporiti njegovu obradu.

Semantička i sintaktička analiza su napredniji postupci obrade koji uključuju analizu n-grama susjednih riječi (npr. bigrama, trigrama) za prepoznavanje obrazaca u jeziku, zatim gramatičku analizu (engl. *POS tagging*) kao dodavanje gramatičkih oznaka kako bi se identificirala funkcija riječi u rečenici te korištenje leksičkih baza podataka poput *WordNeta* za obogaćivanje semantičkih informacija kroz sinonimske, hiperonimske[2] i hijerarhijske odnose između riječi. Također, semantička analiza može uključivati prepoznavanje homonima[3], čime se poboljšava razumijevanje višeznačnih riječi unutar konteksta.

3.2. Vektorske reprezentacije teksta

Model vektorskog prostora (engl. *Vector Space Model*, VSM) matematički je model koji nastaje preslikavanjem riječi ili tekstova u vektore (Salton i sur., 1975; Van Rijsbergen, 1979). To znači da se svakoj riječi i/ili dokumentu dodjeljuje niz brojeva (koordinata, dimenzija) koji predstavljaju njezine karakteristike. Vektori se potom mogu koristiti za

1 N-torke znakova mogu biti korisne za mogućnost rada s rijetkim riječima.
2 Hiperonimi predstavljaju nadređene pojmove (npr. hiperonim riječi „pas” može biti riječ „životinja”).
3 Homonimi su riječi koje imaju isti izraz, ali različit sadržaj (npr. „list” u značenju 'dio biljke' i u značenju 'papir' ima jedan te isti izraz).

izračunavanje sličnosti između dokumenata. VSM predstavlja ključni polazni alat na kome se temelji ovo istraživanje. Dakle, VSM predstavlja tekstne dokumente, rečenice ili riječi kao vektore u višedimenzionalnom prostoru. Sličnost između pojedinih dokumenata korpusa može se izračunavati korištenjem vektora koji predstavljaju dokumente i neku mjeru sličnosti (ili udaljenosti), poput kosinusne sličnosti. Modeli vektorskog prostora ključni su za mnoge NLP zadatke. Njihova sposobnost efikasnog prezentiranja semantičkih veza između riječi i dokumenata koristi se u različitim aspektima obrade prirodnog jezika, poput pretraživanja dokumenata, klasifikacije teksta, grupiranja dokumenata i prepoznavanja entiteta. Postoji nekoliko različitih pristupa za izgradnju VSM-a.

3.2.1. Statistički pristupi

Statistički pristup koristi ili statističke mjere za ponderiranje riječi u dokumentu na temelju njihove učestalosti u tom dokumentu i u cijelom korpusu (TF-IDF) ili statističke metode poput singularne vrijednosne dekompozicije za analizu matrice pojmova i dokumenata, kao i pronalaženje latentnih semantičkih tema među dokumentima (latentno semantičko indeksiranje / latentna semantička analiza, LSI/LSA). Statistički se modeli usredotočuju na opisivanje i generalizaciju podataka. Oni koriste statističke metode za procjenu parametara modela i za testiranje hipoteza o podacima. Cilj je statističkih modela dobivanje uvida u strukturu podataka i prognoziranje budućih podataka ili događaja

Statistički modeli u kontekstu prezentacije teksta koriste matematičke metode kako bi opisali i generalizirali obrasce u jeziku. Pretpostavka je da se značenje riječi može izvesti iz njihove distribucije i odnosa s drugim riječima u velikom korpusu teksta. Kako bi kvantificirali odnose između riječi, statistički modeli koriste statističke metode poput brojenja frekvencije riječi, analize supojavljivanja riječi i modele vjerojatnosti. Cilj statističkih modela koji predstavljaju tekst jest stvoriti takve vektorske reprezentacije riječi ili dokumenata koje čuvaju njihovo značenje i odnose s drugim riječima ili dokumentima. Primjeri takvih statističkih modela su: *vreća riječi* (engl. *Bag-of-Words,* BoW), model koji predstavlja tekst kao neuređeni skup riječi gdje je svaka riječ predstavljena vektorom koji označava njezinu prisutnost ili odsutnost u dokumentu; model *Term Frequency-Inverse Document Frequency* (TF-IDF), koji proširuje BoW tako što uzima u obzir važnost riječi u cijelom korpusu teksta[1], metoda *Latent Semantic Analysis* (LSA) koja koristi dekompoziciju singularnih vrijednosti

1 Što se riječ pojavljuje u više dokumenata, to je ona manje važna.

(engl. *Singular Value Decomposition*, SVD) kako bi otkrila latentne semantičke odnose između riječi i dokumenata; te *modeli tema* (engl. *Topic models*), koji pretpostavljaju da se dokumenti sastoje od mješavine različitih tema, a svaka je tema definirana distribucijom riječi.

3.2.1.1. Pristup na temelju učestalosti riječi u dokumentu i korpusu (TF-IDF)

Jedan je od načina preslikavanja teksta u njegove vektorske reprezentacije i statistički pristup TF-IDF (Bassil i Semaan, 2012; Salton i Buckley, 1988) – tehnika za izračunavanje pondera koji odražava važnost riječi u dokumentu, tehnika koja u konačnici stvara VSM, a koja se realizira matematičkim postupkom u sljedećim koracima:

1. Za svaku riječ u svakom dokumentu broji se frekvencija te riječi u dokumentu (engl. *Term Frequency, TF*).
2. Za svaku se riječ u svakom dokumentu računa koliko je ona relativno važna ili informativna u cijelom korpusu dokumenata. Riječi koje se rijetko pojavljuju u cijelom korpusu informativnije su te stoga važnije. Za svaku se dakle riječ računa inverzna frekvencija dokumenta (engl. *Inverse Document Frequency, IDF*), prema formuli

$$IDF(w) = \log\left(\frac{N}{df(w)}\right) \tag{16}$$

gdje je N broj dokumenata u korpusu, a $df(w)$ broj dokumenata koji sadrže riječ w. Logaritamska funkcija se koristi za smanjenje utjecaja vrlo čestih riječi. *IDF* služi kako bi se težinske vrijednosti riječi prilagodile njihovoj distribuciji u cijelom korpusu, što pomaže u izdvajanju ključnih riječi i smanjenju utjecaja često korištenih riječi

3. Svaki dokument preslikava se u pripadni vektor u kojem svaka dimenzija odgovara jednoj riječi, a vrijednost u toj dimenziji predstavlja težinu *TF-IDF* za tu riječ u tom dokumentu.
4. Nakon što su svi dokumenti predstavljeni svojim vektorima nastala kolekcija vektora predstavlja VSM u kome svaki vektor predstavlja jedan dokument.

3.2.1.2. Pristup matricom pojmova i dokumenata te njezine dekompozicije

VSM može statističkim pristupom nastati i u okviru tzv. latentne semantičke analize, engl. *Latent Semantic Analysis* (LSA), poznate još i pod nazivom *latentno semantičko indeksiranje* – engl. *Latent Semantic Indexing* (LSI). To je tehnika koja se koristi u NLP-u i dohvatu informacija (engl. *Information Retrieval*, IR) kako bi se analizirali odnosi između

skupa dokumenata i pojmova koje ti dokumenti sadrže (Deerwester i sur., 1990; Landauer i sur., 1998). Temelji se na pretpostavci da riječi sa sličnim značenjima imaju tendenciju pojave u sličnim kontekstima unutar tekstova. LSA djeluje na temelju identifikacije uzoraka korištenja riječi u velikim skupovima podataka putem statističkih kalkulacija. Može se reći da se LSA/LSI bavi otkrivanjem skrivene strukture u kolekciji tekstova, pritom koristeći dekompoziciju matrice pojmova i dokumenata kao alatom za prepoznavanje tema, a pozitivna nuspojava je smanjenje dimenzionalnosti i stvaranje VSM-a koji čuva semantičke veze između dokumenata i riječi. VSM se ovom tehnikom dobiva u sljedećim koracima:

1. Izgradnja matrice pojmova i dokumenata čije dimenzije ovise o broju dokumenata i broju pojmova, gdje svaki element matrice predstavlja broj pojave pojma u dokumentu, no može biti i obratno, tj. da su pojmovi u recima, a dokumenti u stupcima. Potonja se varijanta rjeđe koristi jer je prvu varijantu najčešće lakše interpretirati, a nastala matrica je češće manja i rjeđe ju je potrebno komprimirati. Normalizacija matrice je potom neobavezni, ali uobičajeni korak ako se želi smanjiti utjecaj različitih dužina dokumenata ili frekvencija pojmova, a s istim se ciljem može koristiti i neka druga mjera težine poput TF-IDF.
2. Provodi se postupak dekompozicije singularnih vrijednosti (engl. *Singular Value Decomposition*, SVD) matrice pojmova i dokumenata. Ta tehnika linearno transformira matricu u dva vektorska prostora manjih dimenzija, prostora koji su međusobno ortogonalni: prostor dokumenata i prostor pojmova, a postupkom se matrica pojmova i dokumenta razlaže u tri matrice: (a) matrica koncepata dokumenata koja pokazuje koje se teme pojavljuju u svakom dokumentu (obično se označava s U i predstavlja vektore tema dokumenata), (b) dijagonalna matrica singularnih vrijednosti koja pokazuje važnost svake teme (oznaka S) te (c) matrica tema pojmova koja pokazuje koji se pojmovi pojavljuju u svakoj temi (oznaka V, predstavlja vektore tema pojmova).
3. Prethodno proveden postupak SVD omogućava treći korak: smanjenje dimenzije vektorskog prostora, čime se, s jedne strane, zadržavaju najvažnije informacije, s druge strane, smanjuje se šum tj. uklanjaju se manje važne ili nepotrebne informacije, a s treće strane, smanjuje se količina podataka potrebnih za pohranu relativno iste ili usporedive količine informacija. To se postiže odbacivanjem singularnih vektora s niskim singularnim vrijednostima, dok se ostavljaju oni s visokima, koji nose najviše informacija o strukturi podataka. Broj dimenzija na koji se reducira matrica singularnih vrijednosti

ovisi o specifičnom zadatku i dostupnim resursima.

Rezultat LSA postupka je transformirani vektorski prostor, gdje su dokumenti i pojmovi predstavljeni vektorima u niže dimenzionalnom prostoru, čime se olakšava analiza sličnosti među dokumentima i otkrivanje skrivenih semantičkih struktura. Taj se prostor naziva semantički prostor jer bolje odražava semantičke veze između dokumenata i pojmova te omogućuje bolje razumijevanje konteksta.

3.2.2. Probabilistički pristup

Probabilistički modeli su vrsta statističkih modela koji koriste vjerojatnost za opisivanje nesigurnosti u podacima. Oni pretpostavljaju da podaci dolaze iz raspodjele vjerojatnosti i koriste Bayesovu formulu[1] za ažuriranje svojih očekivanja (procjena vjerojatnosti) o parametrima modela na temelju novih podataka (LDA) ili nenegativnu faktorizaciju (engl. *Non-negative Matrix Factorization*, NMF) matrice pojmova i dokumenata na dvije matrice koje predstavljaju dokumente i teme. Cilj je probabilističkih modela dobivanje probabilističke reprezentacije podataka i izračun vjerojatnosti različitih hipoteza o podatcima. Glavna je razlika između statističkih i probabilističkih modela u tome što statistički modeli koriste frekvencije za opisivanje podataka, dok probabilistički modeli koriste vjerojatnosti. Najznačajniji predstavnik probabilističkog pristupa jest stvaranje VSM pomoću metode latentne Dirichletove alokacije (engl. *Latent Dirichlet Allocation*, LDA) (Blei i sur., 2003). Bez obzira na način dobivanja modela VSM iz korpusa dokumenata, takav model numerički prezentira tekst, tj. predstavlja dokumente kao vektore, što drugim metodama omogućava da rade s tekstnim podacima na numerički način. LDA je metoda kojom se automatski otkrivaju teme u korpusu dokumenata inicijalno predstavljenih VSM-om, a s ciljem uspješnijega razumijevanja korpusa dokumenata. LDA može identificirati važnost riječi u odnosu na određenu temu (Blei i sur., 2003). Model LDA može se trenirati na VSM reprezentaciji dokumenata. To omogućava modelu LDA rad s tekstnim podacima na numerički način, što je potrebno za izračunavanje vjerojatnosti.

Tijekom treniranja modela LDA na korpusu dokumenata model uči prepoznati obrasce u podacima i na temelju tih obrazaca konstruira teme (skupine riječi koje se često pojavljuju

1 Bayesova formula uvjetne vjerojatnosti, odnosno vjerojatnosti da se dogodio događaj A ako nam je poznato da se realizirao događaj B, pri čemu su događaji A i B međusobno nezavisni.

$$P(B|A)=\frac{P(B\cap A)}{P(A)} \quad (17)$$

zajedno u dokumentima). Za svaki dokument u korpusu model izračunava vjerojatnost da se taj dokument sastoji od svake teme. Konkretno, za korpus tekstova određuje se broj tema (iskustveno, s obzirom na broj dokumenata, količinu teksta u njima te s obzirom na tematiku), a potom model uči distribuirati riječi u teme i teme u dokumente. Za svaku temu model stvara listu riječi s najvećim vjerojatnostima, što daje uvid u sadržaj i značenje svake teme. Za svaki dokument model stvara distribuciju tema, koja pokazuje vjerojatnost pripadanja toga dokumenta svakoj temi. Teme koje su utvrđene pristupom LDA mogu se koristiti kao nove dimenzije za VSM, pa se svaki dokument može preslikati u vektor u prostoru tema gdje svaka koordinata predstavlja vjerojatnost teme u dokumentu.

3.2.3. Vektorsko reprezentiranje teksta jezičnih modela

Prethodno opisani statistički i probabilistički pristup dobivanju VSM-a povijesno prethode pristupu korištenja vektorskih reprezentacija teksta. Vektorske prostore tim pristupom mogu stvarati tzv. plitke i duboke neuronske mreže (engl. *Shallow Neural Networks,* SNN*, Deep Neural Networks,* DNN). Obje vrste koriste neuronske mreže s jednim (plitke) ili više (duboke) slojeva skrivenih neurona za učenje vektorskih reprezentacija riječi. Pored njih postoji i tzv. duboko učenje (engl. *Deep Learning*, DL), pristup dobivanju VSM-a koji uključuje (nadzirano ili nenadzirano) treniranje (uglavnom dubokih) neuronskih mreža (Goodfellow i sur., 2016). U tom pristupu, umjesto da svaka riječ bude jedna dimenzija vektora, neuronska mreža nauči reprezentirati riječi u gusto distribuiranom vektorskom prostoru (Worth, 2023). Pomalo iznenađujuće za prve istraživače (Mikolov i sur., 2013a) jest skrivena semantika takva VSM-a, činjenica da slične riječi imaju slične vektore, tj. u vektorskom prostoru nalaze se utoliko bliže jedan drugome koliko su međusobno semantički bliži. Postoje različite tehnike učenja vektora koji se koriste za taj pristup, poput *Word2Vec*, *Doc2Vec*, *FastText*, *GloVe*, *USE*, *ELMo*, *BERT*, *Laser Embeddings* i mnogi drugi veliki jezični modeli (engl. *Large Language Models*, LLMs). Plitke i duboke neuronske mreže u pravilu su brže i jednostavnije za treniranje od modela dubokog učenja i mogu se koristiti s manjim skupovima podataka, no ne postižu istu razinu točnosti kao modeli DL te se lošije ponašaju s novim podacima. Ove neuronske mreže prethodile su modelima dubokog učenja, ali su potonji postali dominantni. Dakle, u obradi prirodnog jezika, duboko učenje je skup tehnika za treniranje i korištenje dubokih neuronskih mreža (iznimno i plitkih) kao arhitekture neuronske mreže, skup tehnika koji se koristi za dobivanje vektorskih reprezentacija riječi.

DNN predstavljaju alat koji se koristi u dubokom učenju, ali duboko učenje uključuje i niz drugih elemenata, kao što su algoritmi optimizacije i funkcije gubitka[1] (engl. *loss function*).

Duboko učenje je područje umjetne inteligencije koje se fokusira na učenje velikih višeslojnih neuronskih mreža. Te neuronske mreže, inspirirane strukturom ljudskog mozga, mogu učiti iz golemih skupova podataka i izvoditi složene zadatke. Dakle, DL neuronske mreže sastoje se od više slojeva koji obrađuju podatke na složen način. Svaki sloj preuzima informacije iz prethodnog sloja i transformira ih na način koji omogućava modelu da otkrije sve kompleksnije karakteristike unutar podataka (Goodfellow i sur., 2016). Duboke neuronske mreže automatski uče svojstva podataka tijekom procesa treniranja, pri čemu koriste sofisticirane algoritme optimizacije koji ažuriraju parametre modela tijekom učenja, s ciljem smanjenja pogrešaka u kasnijem korištenju modela i poboljšanja njegovih performansi (Zhang i sur., 2023). Naučena svojstva preslikavaju se tijekom učenja u parametre višedimenzionalnih VSM-a koji potom omogućavaju modelu da bolje „reagira" ili „generalizira" na nove, prethodno neviđene podatke (Goodfellow i sur., 2016). S obzirom na velik broj parametara koji DL neuronske mreže trebaju naučiti, prilikom učenja je potrebna i velika količina podataka kako bi se omogućilo modelu generaliziranje s obzirom na nove podatke (Mikolov i sur., 2013b). U suprotnom može se dogoditi prenaučenost (engl. *overfitting*) i nezadovoljavajuća ekstrapolacija modela na nove podatke, odnosno nedostatna sposobnost prilagođavanja modela novim podacima.

Ako statističke i probabilističke pristupe okarakteriziramo kao „klasični pristup", a vektorsko reprezentiranje riječi kao „noviji i moderniji pristup", treba reći da oba pristupa imaju svoje prednosti i nedostatke te se često koriste u različitim kontekstima, ovisno o zadatku i dostupnosti podataka. Primjerice, TF-IDF je jednostavan i interpretativan, ali rezultirajući VSM ne zadržava kompleksne semantičke odnose između riječi. S druge strane, vektorska reprezentacija pruža bogatije semantičko predstavljanje riječi, ali, s jedne strane, zahtijeva velike količine podataka za treniranje, a, s druge strane, dobiveni VSM ne može se uvijek lako interpretirati. U istraživanju su korišteni *Word2Vec* (podmodeli *Continuous Bag of Words*, *CBoW* i *Skip-Gram*), *Doc2Vec* (podmodeli *Distributed Bag of Words*, DBoW i *Distributed Memory*, DM), *FastText*, *Glove*, *USE*, *ELMo*, *BERT*, *Laser Embeddings* te danas najviše korišteni predtrenirani veliki jezični modeli koji mijenjaju način korištenja

1 Funkcija gubitka koristi se prilikom treniranja modela dubokog učenja. Pomoću nje se usmjerava proces optimizacije kako bi model postao što precizniji tijekom treniranja. Zapravo je cilj treniranja modela minimizirati vrijednost funkcije gubitka, smanjiti razliku između predviđenih i stvarnih (ciljanih) rezultata.

informacijsko-komunikacijske tehnologije. To su modeli strojnog učenja trenirani na velikim količinama teksta (reda veličine TB), što im omogućuje izvršavanje kompleksnih zadataka simulirajući ljudsko izražavanje: prevođenje s jezika na jezik, generiranje teksta i odgovaranje na upite. Veliki jezični modeli ipak, s obzirom na to da su trenirani kontroliranim ulaznim podacima, povremeno haluciniraju (Xu i sur., 2024) i mogu biti pristrani (Yeh i sur., 2023), već prema tome s kojim su ih podacima njihovi tvorci stvarali. Svi navedeni i u eksperimentima korišteni modeli osim *GloVe*[1] i *FastText*[2] modela, su (veliki) jezični modeli temeljeni na dubokom učenju.

Iako se 2013. smatra značajnom godinom za DL, tj. pojavom radova o istraživanjima Mikolova, povijest DL počinje ranije. Tako 1940-ih godina počinju istraživanja neuronskih mreža – temelja za DL: McCulloch i Pitts (1943) predlažu model neurona koji može izvršiti jednostavne računske operacije (McCulloch i Pitts, 1943). 1960-ih su razvijene prve višeslojne neuronske mreže, no ograničena računalna snaga toga vremena otežala je njihovu praktičnu primjenu (Eliseev i Sknarina, 2012). 1980-tih jača zanimanje za istraživanje neuronskih mreža zbog razvoja algoritma *backpropagation*, koji omogućava efikasno učenje u višeslojnim mrežama (Rumelhart i sur., 1986). 2000-ih modeli DL postaju uspješni u prepoznavanju rukopisa, ali je primjena modela DL i dalje vrlo ograničena s obzirom na nedostatak računalne snage i velikih skupova podataka. Pravu (r)evoluciju razvoja modeli DL doživljavaju od 2012. godine. Naime, događa se značajan napredak u računalnoj snazi i u dostupnosti velikih skupova podataka, što omogućava modelima DL postizanje vrhunske učinkovitosti na širokom spektru zadataka. 2013. godine objavljen je rad koji je popularizirao koncept *Word2Vec* modela za učenje vektorskih reprezentacija riječi (Mikolov i sur., 2013b). Konačno, DL model *AlexNet*, temeljen na konvolucijskim neuronskim mrežama, postigao je revolucionarne rezultate na zadatku prepoznavanja slika na *ImageNet* natjecanju (Krizhevsky i sur., 2017).

3.2.3.1. Word2Vec

Word2Vec je pristup obradi prirodnog jezika potekao iz *Googlea*, koji se koristi za dobivanje vektorskih reprezentacija riječi, s ciljem utvrđivanja odnosa između riječi iz velikog korpusa teksta. Ti vektori obuhvaćaju informacije o značenju riječi na temelju okolnih riječi. Word2Vec algoritam stvara reprezentacije teksta u vektorskom prostoru, učenjem

1 *GloVe* model je model matrične faktorizacije.
2 *FastText* se oslanja na tablicu pretraživanja/sučeljavanja i statističke metode za učenje vektora riječi.

neuronske mreže s velikim tekstnim korpusom. Model Word2Vec nije jedan algoritam, to je skupina srodnih arhitektura i optimizacija koje se mogu koristiti za učenje vektorskih reprezentacija riječi iz velikih skupova podataka (Mikolov i sur., 2013b). Vektori naučeni kroz Word2Vec pokazali su se uspješnima u nizu zadataka obrade prirodnog jezika (Mikolov i sur., 2013a).

Iako se Word2Vec klasificira kao DL model, umjesto DNN on koristi SNN. Razlog za to su povijesni kontekst i evolucija dubokog učenja. U ranoj fazi razvoja, DL se odnosio na neuronske mreže s više skrivenih slojeva. S vremenom definicija se proširila i obuhvatila modele koji koriste sofisticirane algoritme treniranja i optimizacije, čak i ako imaju relativno mali broj slojeva. Word2Vec uklapa se u tu proširenu definiciju dubokog učenja i često se opisuje kao plitka neuronska mreža, čime se naglašava njegova jednostavnija struktura. Koristi algoritam *backpropagation* za optimizaciju težina neuronske mreže te je, iako najstariji DL model iz novijega vremena, i nadalje vrlo uspješan u raznim NLP zadacima. Word2Vec temelji se na plitkim neuronskim mrežama s dva sloja, koje se treniraju kako bi rekonstruirale lingvističke kontekste riječi (Mikolov i sur., 2013b). Word2Vec ima relativno jednostavnu arhitekturu u usporedbi s modernim modelima dubokog učenja koji imaju stotine ili tisuće slojeva. Postoje dva glavna podmodela Word2Veca: *Continuous Bag of Words*, CBoW i *Skip-Gram,* SG. Prvi pokušava prognozirati središnju riječ iz okolnih riječi u određenom prozoru riječi, dok potonji pokušava predvidjeti okolne riječi za zadanu (središnju) riječ.

Ako je dobro uvježban, Word2Vec je u stanju identificirati sinonime ili dati prijedloge riječi za nepotpunu rečenicu. Za istu riječ u različitim kontekstima daje isti vektor, ali ga ne može dati za riječi koje nisu bile u ulaznom korpusu kojim se model trenirao.

3.2.3.2. Doc2Vec

Word2Vec izvrstan je kod vektorskih reprezentacija riječi, no nije dizajniran za generiranje jedinstvene reprezentacije višestrukih riječi, rečenica, paragrafa ili dokumenata. Mogući pristup tome problemu jest dobivanje vektora za svaku riječ, a potom izračun prosjeka (ili zbroja) vektora riječi za složenije strukture. Nažalost, takav pristup ne daje tako dobre rezultate kao neki drugi modeli DL. Le i Mikolov (2014) uvidjeli su potrebu za vektorskom reprezentacijom složenijih tekstnih struktura te su predložili Doc2Vec model. (Le i Mikolov, 2014). Njime su ponudili rješenje za model pretvorbe varijabilnog broja riječi u

vektorski oblik fiksne veličine. Ovdje se u biti i dalje koristi Word2Vec model, ali mu je dodan vektor *Paragraph ID* kao proširenje CBOW modela iz Word2Veca. Dok se neuronska mreža trenira vektorima riječi, pritom se ujedno stvaraju i vektori dokumenta. Taj se model zove *Distributed Memory version of Paragraph Vector*, PV-DM. Model pamti kojem kontekstu pripadaju riječi koje nadolaze, poput naslova odlomka. Dakle, vektori riječi prezentiraju riječi, a vektori dokumenata prezentiraju dokumente. Kao što u modelu Word2Vec postoje dva podmodela, CBOW i Skip-Gram, tako i u modelu Doc2Vec pored PV-DM postoji i podmodel koji se naziva *Distributed Bag of Words version of Paragraph Vector*, PV-DBoW, a pandan je Skip-gramu iz Word2Veca. PV-DBoW je brži i zahtijeva manje memorije od Word2Vecovog Skip-grama jer nema potrebe pohranjivati vektore riječi. DBoW se fokusira na predviđanje riječi u dokumentu iz vektora dokumenta, a DM se fokusira na predviđanje konteksta riječi (okolnih riječi) iz središnje riječi i vektora dokumenta. Kombiniranjem tih dvaju modela *Doc2Vec* postiže bolje rezultate u usporedbi s modelima koji se fokusiraju samo na dokumente ili samo na riječi.

3.2.3.3. GloVe

Dvije glavne vrste metoda za učenje vektora riječi jesu: metode faktorizacije globalne matrice, kao što je latentna semantička analiza (LSA), i modeli lokalnoga kontekstnog prozora poput *Word2Vec*. Prve metode učinkovito koriste statističke podatke, ali slabo pronalaze analogiju između riječi, što ukazuje na suboptimalnu strukturu vektorskih prostora koje stvaraju. Druge vrste, modeli poput Skip-Grama iz modela Word2Vec, uspješno otkrivaju analogije između riječi, ali slabo koriste statistiku korpusa jer uče na zasebnim lokalnim kontekstnim prozorima umjesto na globalnim podacima iz matrice koincidencije. Prema Pennington i sur. (2014), model Word2Vec otkrio je semantičke i sintaktičke pravilnosti, ali izvor tih pravilnosti nije dovoljno razjašnjen (Pennington i sur., 2014). Oni su stoga predstavili model GloVe (engl. *Global Vectors for Word Representation*), hibridni model dubokog učenja i reprezentacije korištenjem matrične faktorizacije koji iskorištava prednosti obaju pristupa. Model efikasno iskorištava statističke podatke (nenadziranim) učenjem isključivo na onim elementima matrice riječ-riječ (engl. *word-word cooccurrence matrix*) koji nisu nule, a ne na cijeloj rijetkoj (engl. *sparse*) matrici ili na kontekstnim prozorima velikoga korpusa, te stvara VSM koji ima smislenu podstrukturu. Cilj je treniranja modela *GloVe* dobivanje takvih vektora da je njihov skalarni produkt jednak logaritmu vjerojatnosti riječi iz

matrice kolokacije. Razlika između vektora u VSM ovdje je jednaka razlici logaritama vjerojatnosti pojave riječi u kolokaciji. S obzirom na to da omjeri vjerojatnosti imaju značenje, ta je informacija ukdirana u razlike vektora kao razlike logaritama vjerojatnosti tj. logaritam omjera vjerojatnosti. GloVe model trenira se na velikoj količini teksta, no poput Word2Veca, nema vektorskog rješenja za riječi koje nisu bile u tekstnom korpusu tijekom učenja. Model GloVe u smislu rječnika i pripadnih vektora riječi dostupan je za besplatno preuzimanje, što je predstavljalo poticaj daljnjim istraživanjima.

3.2.3.4. FastText

Bojanowski i sur. (2016) uočili su da prethodni vektorski prikazi zanemaruju morfologiju riječi, dodjeljujući različitim riječima novi vektor, što predstavlja ograničenje, osobito za flektivne i morfološki bogate jezike koji se odlikuju velikim rječnicima i velikim brojem rijetkih riječi (Bojanowski i sur. 2016). Njihov model *FastText*, poput Word2Veca, koristi kontinuirani prikaz riječi, trenira se na velikim neoznačenim korpusima, ali s tom razlikom što se ovdje svaka riječ predstavlja kao vreća (engl. *Bag*) n-grama znakova. Vektorski prikaz povezan je sa svakim znakom n-grama, a riječi su predstavljene zbrojem vektorskih prikaza n-grama znakova. *FastText* se ne smatra punopravnim DL modelom jer ne koristi slojevitu neuronsku mrežu za učenje reprezentacija riječi. Umjesto toga *FastText* koristi jednostavniji algoritam koji se temelji na tablici pretraživanja (Joulin i sur., 2016). Tablica pretraživanja (engl. *lookup table*) predstavlja način pohranjivanja vektorskih reprezentacija riječi. Umjesto treniranja neuronske mreže *FastText* koristi statističke metode za izračunavanje vektorskih reprezentacija riječi na temelju n-grama riječi. Za razliku od Word2Veca, koji se fokusira samo na cijele riječi, *FastText* uzima u obzir i informacije o dijelovima riječi. Razbija riječi na n-grame znakova (sekvence od n znakova). Ti n-grami čuvaju morfološke sličnosti i omogućuju modelu razumijevanje strukture riječi. Tijekom treniranja *FastText* analizira veliki korpus teksta i računa koliko se često n-grami pojavljuju s drugim riječima u kontekstu. Te informacije o supojavljivanju pomažu modelu da razumije semantiku riječi. Riječi koje se pojavljuju u sličnim kontekstima vjerojatno će imati slična značenja. Svaka riječ i n-gram u rječniku dobivaju nasumični početni vektor. *FastText* potom koristi arhitekturu Skip-gram ili CBOW (sličnu Word2Vecu) za treniranje. Ovisno o odabranoj arhitekturi, model predviđa okolne riječi (Skip-gram) ili središnju riječ (CBOW) na temelju danoga konteksta n-grama. Na temelju točnosti predviđanja model ažurira vektore

uključenih riječi i n-grama. Riječi s istim kontekstima imat će svoje vektore bliže u vektorskom prostoru, što odražava njihovu semantičku sličnost. Proces predviđanja i ažuriranja vektora obavlja se tijekom mnoštva iteracija nad korpusom. Tijekom vremena/iteracija vektori u tablici pretraživanja se usavršavaju, sve više odražavajući semantičke veze između riječi na temelju obrazaca supojavljivanja. Svojom koncepcijom korištenja n-grama *FastText* uspješno daje vektorske reprezentacije i onih riječi koje nisu bile u korpusu za učenje.

3.2.3.5. USE

Transformer je arhitektura neuronske mreže koja je uvela tzv. mehanizam pažnje (engl. *attention*), koji modelu omogućuje fokusiranje na najrelevantnije dijelove rečenice prilikom obrade teksta (korištenjem ponderiranja kojim dodjeljuje veće težine relevantnijim dijelovima teksta), što je posljedično dovelo do poboljšanja performansi raznih NLP zadataka (Vaswani i sur., 2017). Arhitektura transformera sastoji se od dviju komponenata: enkodera i dekodera. Enkoder obrađuje ulazni tekst i generira reprezentaciju konteksta za svaku riječ u rečenici, a dekoder koristi tu reprezentaciju konteksta za generiranje izlaznoga teksta, riječ po riječ. Model se nadziranim učenjem trenira na velikom korpusu teksta s više milijarda riječi.

USE model (engl. *Unstructured Semantic Embeddings*) je Googleov model otvorenoga koda koji se temelji na arhitekturi transformera, tj. koristi ga za proces učenja semantičkih reprezentacija riječi iz neobrađenih tekstova (Cer, Yang, Kong, Hua, Limtiaco, St. John, i sur., 2018). Transformerov model pažnje (Vaswani i sur., 2017) omogućuje USE modelu učenje dugoročnih ovisnosti između riječi u rečenici, što je važno za učenje semantičkih reprezentacija riječi, a s obzirom na to da značenje riječi često ovisi o kontekstu u kojem se ona nalazi. USE model uči reprezentacije riječi iz neobrađenih tekstova, čineći ga pogodnijim za zadatke kao što su strojno prevođenje, sažimanje teksta i odgovaranje na pitanja. Model preslikava proizvoljno dugačak tekst u vektorski prikaz fiksne dužine.

3.2.3.6. ELMo

ELMo (engl. *Embeddings from Language Models*) je model istraživača iz *Allen Institute for Artificial Intelligence* (AI2) iz 2018. godine koji koristi dvosmjerni LSTM (dugoročno-kratkoročnu memoriju) model za stvaranje reprezentacija riječi (Peters i sur., 2018). Umjesto reprezentacija riječi modela poput *Word2Vec* ili *GloVe* koji dodjeljuju fiksni vektor svakoj riječi bez obzira na kontekst, *ELMo* generira reprezentacije riječi temeljene na

kontekstu cijele rečenice. Stoga model stvara više vektorskih reprezentacija za svaku riječ, pri čemu svaka predstavlja drukčiji aspekt njezina značenja na temelju okolnih riječi. Model se sastoji od nekoliko slojeva neuronskih mreža na razini znakova (CNN) te dva sloja dvosmjernih LSTM. Ti slojevi generiraju kontekstno prilagođene reprezentacije riječi koje se zatim koriste za računanje reprezentacija za zadatke poput analize sentimenta, prepoznavanja imenskih entiteta i odgovaranja na pitanja jer može uhvatiti različita značenja riječi, ovisno o kontekstu rečenice. To ga čini posebno korisnim u zadacima u kojima kontekst igra ključnu ulogu, kao što su razumijevanje i generiranje prirodnoga jezika

3.2.3.7. Laser Embeddings

Laser Embeddings još je jedna tehnika za stvaranje višejezičnih vektorskih reprezentacija rečenica koja se temelji se na arhitekturi transformera (Artetxe i Schwenk, 2019). U stanju je predstavljati rečenice iz različitih jezika na način koji obuhvaća njihovo značenje, bez obzira na korišteni jezik. Osnovna je ideja da rečenice sa sličnim značenjima trebaju biti bliske jedna drugoj u VSM, čak i ako su napisane na različitim jezicima. Predviđen je za korištenje pri prevođenju, pretraživanju dokumenata, prepoznavanju semantičkih sličnosti, pretraživanju rečenica, svugdje gdje je potrebno usporediti značenja rečenica preko različitih jezika. Temelji se na dvosmjernoj LSTM arhitekturi s mehanizmom pažnje. Laser Embeddings je 2018. godine predstavio *Facebook AI Research* tim (Artetxe i Schwenk, 2019) i objavio ga kao projekt otvorenoga koda.

3.3. Veliki jezični modeli

U poglavlju *2.4. Metode akademskog plagiranja* definirani su jezični modeli kao klasa algoritama za obradu prirodnog jezika sa sposobnošću predviđanja sljedeće riječi u nizu (Nandakumar i sur., 2023; Radford i sur., 2019). Pored statističkih metoda kojima se jezični modeli koriste za izračunavanje vjerojatnosti pojave određene riječi u danom kontekstu, za obradu informacija koriste i arhitekture neuronskih mreža poput transformera. Te se arhitekture treniraju na velikim skupovima podataka kako bi naučile kompleksna obilježja i reprezentacije jezika. Jezični modeli (engl. *language model*, LM) kombiniraju elemente statistike i dubokog učenja za postizanje svojih ciljeva. U posljednjih nekoliko godina jezični modeli postali su sve moćniji zahvaljujući dostupnosti većih skupova podataka, snažnijega računalnog hardvera te sve boljih algoritama i arhitektura dubokog učenja.

Veliki jezični modeli (engl. *large language models*, LLM) su vrsta modela dubokog učenja koji su posebno dizajnirani za obradu i razumijevanje prirodnog jezika. LLM-ovi se temelje na neuronskim mrežama treniranima na velikim količinama tekstova – knjigama i člancima te izvornom programskom kodu (Birhane i sur., 2023; Naveed i sur., 2024; Sundaresan, 2022), što im omogućuje generiranje teksta, prevođenje jezika, pisanje različitih vrsta kreativnih sadržaja i odgovaranje na pitanja. Neuronske mreže LLM-ova uče prepoznati obrasce u podacima i predvidjeti sljedeću riječ ili rečenicu. Što je većoj količini podataka izložen neki LLM tijekom učenja, to će model postati bolji u generiranju teksta koji je sličan tekstu na kojem je učen. LLM-ovi su sposobni obavljati mnoge zadatke koji su nekoć bili smatrani isključivom domenom ljudi: generiranje teksta, prevođenje i odgovaranje na pitanja. Ipak, unatoč svemu, mogu biti pristrani ili generirati tekst koji nije činjenično točan (Brundage i sur., 2018).

LLM-ovi se mogu koristiti za stvaranje vektorskih reprezentacija teksta te postaju sve popularniji zbog svoje sposobnosti čuvanja konteksta i semantike rečenice znatno bolje nego što to mogu tradicionalni modeli poput USE ili ELMo. Konceptualno, LLM-ovi stvaraju vektorske reprezentacije rečenica sljedećim koracima:

1. Tokenizacija: rečenica se dijeli na manje jedinice (riječi ili njezini dijelovi).
2. Ugrađivanje riječi: svaka riječ (ili njezin dio) pretvara se u vektor koji predstavlja njezino značenje.
3. Kontekstualizacija: LLM obrađuje rečenicu i prilagođava vektorske reprezentacije svake riječi na temelju njezina konteksta u rečenici.
4. Agregacija: vektorske reprezentacije riječi se kombiniraju (npr. zbrajanjem ili prosjekom) kako bi se dobio vektor koji predstavlja dio ili cjelinu teksta.

Prednosti pristupa putem LLM vektorskih reprezentacija ogledaju se u tome što LLM-ovi mogu „razumjeti" ili preciznije preslikavati, složenije jezične konstrukcije i nijanse značenja, što rezultira kvalitetnijim vektorskim reprezentacijama i u tome što se mogu dodatno prilagoditi (engl. *fine-tuning*) specifičnim domenama ili zadacima, što dodatno poboljšava njihovu učinkovitost. No korištenje LLM-ova za dobivanje vektorskih reprezentacija riječi i/ili rečenica znatno je zahtjevnije prema računalnim resursima od tradicionalnih modela poput USE. Dodajmo da su LLM-ovi obično veći modeli, što dodatno otežava njihovu implementaciju i korištenje. Stoga se koriste biblioteke poput *Sentence Transformers*, koja pruža jednostavan pristup različitim modelima za stvaranje vektorskih reprezentacija rečenica,

uključujući modele temeljene na LLM-ovima, kao što su BERT i RoBERTa. Drugi pristup je korištenje API-a koji su na raspolaganju za neke LLM modele, tj. pomoću kojih se mogu generirati vektorske reprezentacije tekstnih jedinica.

Veliki jezični modeli stvorili su oko sebe pozamašan ekosustav, u smislu arhitektura i tehnologija koje koriste te teorijskih koncepata, spoznaja, metodologija i zakona do kojih su došli istraživači i znanstvenici potaknuti razvojem LLM-ova (naprimjer otkrićem i definiranjem zakona skaliranja ili neočekivanih sposobnosti LLM-ova) ili su bili preduvjet za njihov razvoj (poput arhitekture transformera).

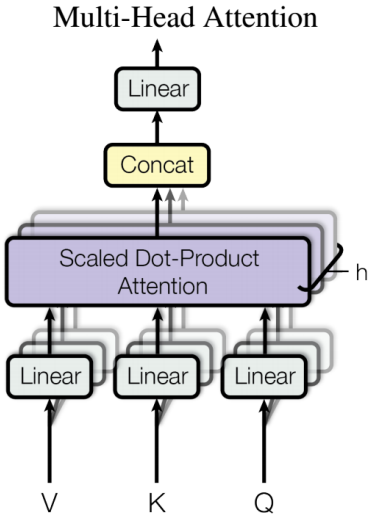
3.3.1. Arhitektura transformera

Google Transformer (Vaswani i sur., 2017) je arhitektura dubokog učenja koja je značajno poboljšala performanse u raznim zadacima obrade prirodnog jezika, poput strojnog prevođenja, sažimanja teksta, označavanja entiteta te drugih sekvencijalnih zadataka (Devlin i sur., 2018; Radford i sur., 2019; Vaswani i sur., 2017). Transformer je ključan za izgradnju velikih jezičnih modela, poput BERT-a i GPT-a koji su revolucionirali primjenu dubokog učenja u NLP-u. Transformer je tip neuronske mreže koji je posebno dizajniran za obradu sekvencijalnih podataka u koje spada primjerice tekst. Ključni mehanizam transformera je višeglava pažnja (engl. *multi-head attention*)[1] prikazana na slici 7 (Vaswani i sur., 2017), koja omogućuje modelu simultano razmatranje različitih aspekata odnosa između riječi. Svaka „glava“ pažnje u modelu obavlja funkciju samopozornosti uzimajući u obzir sve riječi u ulaznoj sekvenci i pritom računa ponderirane vrijednosti za svaku riječ, ovisno o njezinu odnosu s drugim riječima. Opisani mehanizam omogućuje modelu bolje razumijevanje konteksta i veze među udaljenim riječima unutar rečenice te mu omogućuje učenje složenih odnosa između riječi u rečenici, uzimajući u obzir kontekst u kojem se svaka riječ pojavljuje. Značajno je to poboljšanje u odnosu na RNN-e koji se bore s dugoročnim ovisnostima u sekvencama[2], tj. sa zadacima koji zahtijevaju učenje dugoročnih ovisnosti (Hochreiter, 1998),

1 Izrazom se želi naglasiti važnost konteksta u kojem se svaka riječ pojavljuje u mehanizmu.

2 Rekurzivne neuronske mreže (RNN) se bore s dugoročnim ovisnostima u sekvencijama zbog problema s nestajanjem gradijenta (Goodfellow i sur., 2016). Gradijenti funkcije gubitka u odnosu na parametre mreže postaju sve manji kako se ide dublje u mrežu, što mreži otežava učenje dugoročnih ovisnosti, jer signali iz ranijih dijelova sekvence s vremenom slabe. Postoje dva razloga za taj problem. Prvi je dio problema u tome što sigmoidna aktivacijska funkcija koju RNN-i obično koriste ima tendenciju zasićenja za velike vrijednosti ulaza – signali iz ranijih dijelova sekvence vremenom mogu doći na konstantnu vrijednost, što mreži otežava učenje. Drugi dio problema slabljenja gradijenta funkcije gubitka je u tome što se gradijenti mogu multiplicirati s vrijednostima aktivacijskih funkcija u svim slojevima mreže. Ako su vrijednosti aktivacijskih funkcija male, gradijenti će postati sve manji i manji kako se ide dublje u mrežu.

kao što su strojno prevođenje i prepoznavanje govora. Stoga je ta arhitektura, prikazana na slici 8 (Vaswani i sur., 2017), zamijenila tradicionalne rekurzivne neuronske mreže (RNN) u mnogim najsuvremenijim (velikim) jezičnim modelima.

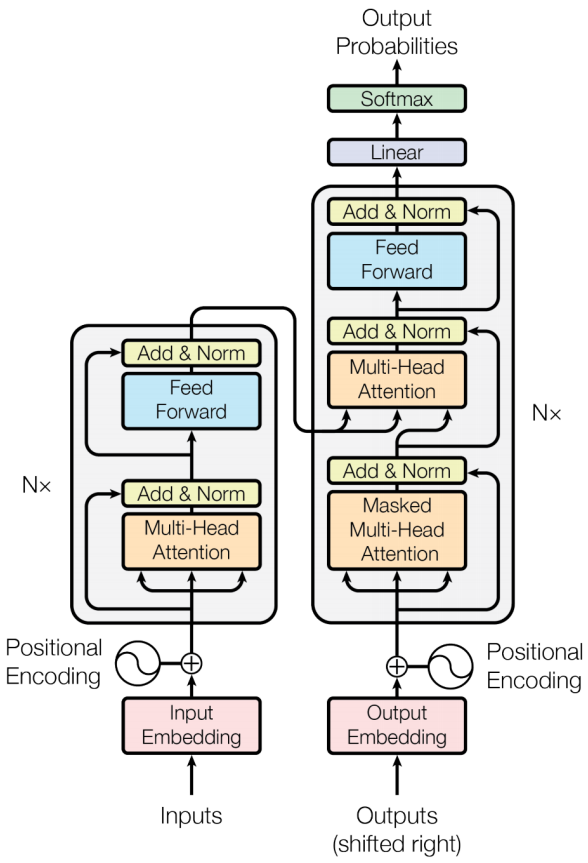


Slika 7. Mehanizam kontekstne pažnje arhitekture transformera (Vaswani i sur., 2017)

Jedan od glavnih doprinosa mehanizma samopozornosti je da za svaku riječ x_i u ulaznoj sekvenci, računa pozornost prema svim drugim riječima x_j u sekvenci, što se može formulirati kao

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{18}$$

gdje su Q (engl. *queries*), K (engl. *keys*) i V (engl. *values*) matrice koje predstavljaju transformirane ulazne riječi, a d_k je dimenzija modela. Mehanizam višeglave pažnje omogućuje da se više takvih operacija odvija paralelno, što je ključ za hvatanje različitih vrsta relacija u tekstu, i to je glavni razlog (Vaswani i sur., 2017) što je ova arhitektura zamijenila tradicionalne rekurzivne neuronske mreže (RNN) u mnogim najsuvremenijim (velikim) jezičnim modelima.



Slika 8. Model arhitekture transformera
(Ménard, 2021; Meuschke i Gipp, 2013; Vaswani i sur., 2017)

Arhitektura transformera sastoji se od dva glavna dijela: enkodera i dekodera. Enkoder uzima ulaznu sekvencu, aplicira mehanizam pozornosti na riječi te generira kontekstualizirane reprezentacije za svaku riječ. Te reprezentacije zatim koristi dekoder za predviđanje izlazne sekvence. Transformeri koriste tzv. pozivnu strukturu koja omogućuje modelu učinkovitu obradu dugoročnih ovisnosti te uklanja potrebu za rekurzijom, što rezultira bržim treniranjem i inferencijom. Vrlo su dobro prilagođeni paralelizaciji tj. sposobni su simultano obrađivati sve elemente unutar sekvenci umjesto serijske obrade kakvu koriste modeli RNN. Transformeri su također sposobni koristiti distribuirane računalne sustave kod treniranja modela. Sve te sposobnosti transformera omogućavaju bržu obradu i skalabilnost (u smislu dodavanja resursa potrebnih za treniranje), što je ključno za treniranje velikih modela na ogromnim količinama podataka. Arhitektura transformera vrlo je skalabilna i u smislu da se može koristiti za izgradnju modela s velikim brojem parametara neuronske mreže (više parametara na ulazu neurona, tj. po jednu težinu za svaku ulaznu vezu te za svaki izlaz

neurona ima pomak), što je važno za postizanje visokih performansi na različitim zadacima NLP-a.

3.3.2. BERT

Google BERT model (engl. *Bidirectional Encoder Representations from Transformers*) je model dubokog učenja koji je treniran na velikoj količini tekstnih podataka, čime se postiže duboko razumijevanje semantičke strukture jezika, a razvio ga je Google AI (Devlin i sur., 2018). BERT se temelji na modelu transformera koji „razumije" kontekst riječi u rečenici (Vaswani i sur., 2017). Zahvaljujući dvosmjernoj obradi riječi (engl. *bidirectional*) model BERT može učinkovito učiti značenje riječi na temelju okolnoga teksta. Za modeliranje dugoročnih ovisnosti između riječi u rečenici umjesto rekurzivnih veza koristi se mehanizam kontekstne pažnje arhitekture transformera. To omogućuje modelu BERT učinkovitu obradu dužih sljedova teksta i bolje razumijevanje kontekstnih veza između riječi. Ključna inovacija BERT-a u tome je što je on unaprijed treniran na dva zadatka. Prvi zadatak je maskirano modeliranje jezika u kome se model trenira tako da može predvidjeti riječi koje su maskirane u tekstu. To prisiljava model na duboko razumijevanje konteksta riječi i njihovu ulogu u rečenici. Drugi zadatak je predviđanje sljedeće rečenice u tekstu, što pomaže modelu razvijanje dugoročnih međuovisnosti riječi, izraza i rečenica u tekstu. Kombinirano, dva navedena zadatka omogućuju BERT-u učenje dubokih značajki i reprezentacija jezika. BERT je značajno poboljšao rezultate na mnogim zadacima NLP-a, i to postavljajući nove standarde za performanse raznih zadataka NLP-a, poput strojnoga prevođenja, računalnih sugovornika (engl. *chatbot*) i analize teksta. BERT je inspirirao razvoj alata i resursa otvorenoga koda koji olakšavaju istraživačima prilagodbu i korištenje modela za svoje potrebe (specifični zadaci, drugi jezici pored engleskoga, različite domene), razvoj brojnih novih modela temeljenih na arhitekturi transformera, te je potaknuo nova istraživanja u području unaprijed treniranih velikih jezičnih modela, što je dovelo do razvoja još moćnijih LLM-ova. Razvijeno je mnoštvo derivata BERT-a prilagođenih specifičnim zahtjevima i korisničkim slučajevima (Feng i sur., 2020; Jagtap, 2020; Jiao i sur., 2020; Joshi i sur., 2019; Lan i sur., 2020; Liu i sur., 2019; Reimers i Gurevych, 2019; Sanh i sur., 2020).

3.3.3. Osnovna obilježja velikih jezičnih modela

U ovome poglavlju dan je kratak pregled osnovnih obilježja velikih jezičnih modela. Glavni cilj bio je prikazati dimenzije vektorskih reprezentacija koje koriste veliki jezični modeli jer će se poslije, u poglavlju *5. Eksperimenti*, prikazati rezultati eksperimenata vezanih uz dimenzionalnost modela. Kako je vidljivo iz tablice 2, u kojoj su prikazani usporedni podaci o navedenim obilježjima, dimenzije vektorskih reprezentacija velikih jezičnih modela imale su trend rasta, no u posljednje vrijeme programeri smanjuju taj broj, što je primjerice vidljivo s OpenAI GPT modelima, gdje je dimenzija vektora u vektorskim reprezentacijama smanjena sa 768-12288 (GPT-3) na 1536-3072 (GPT-4) (OpenAI, 2024b). To je znak da su barem neke velike tvrtke, predvodnice u razvoju velikih jezičnih modela, došle do zaključka da je povećanje broja dimenzija neisplativo i/ili kontraproduktivno.

Tablica 2. Osnovna obilježja LLM-ova

Naziv	God.	Dimenzija VP	Broj parametara	Građa	Otvorenost
GPT-1	2018	256	$1.17 \cdot 10^{11}$	hib	nedostupan
BERT	2018	768-2048	$1.1\text{-}3.4 \cdot 10^{11}$	hib	otvoren
GPT-2	2019	768-1600	$1.17 \cdot 10^{8}\text{-}1.5 \cdot 10^{9}$	mon	zatvoren
T5	2020	512-4096	$6 \cdot 10^{7}\text{-}1.1 \cdot 10^{10}$	hib	otvoren
GPT-3	2020	768-12288	$1.25 \cdot 10^{8}\text{-}1.75 \cdot 10^{11}$	mon	zatvoren, API
Codex	2021	12288	$1.2 \cdot 10^{10}$	mon	zatvoren
Switch Transformers	2022	768 ili 1024	$1.6 \cdot 10^{9}\text{-}1.6 \cdot 10^{18}$	hib	otvoren
T0	2022	1024-4096	$1.5 \cdot 10^{9}\text{-}1.1 \cdot 10^{10}$	mon	otvoren
GLaM	2022	768-8192	$3.8 \cdot 10^{7}\text{-}1.0 \cdot 10^{12}$	mod	nedostupan
WebGPT	2022	768-12288	$1.75 \cdot 10^{11}$	hib	nedostupan
Retro	2022	1024	$7 \cdot 10^{9}$	hib	zatvoren
Nvidia Retro 48B	2022	8192	$4.8 \cdot 10^{10}$	hib	zatvoren
Gopher	2022	4096	$2.8 \cdot 10^{11}$	mon	zatvoren
LaMDA-PT	2022	8192	$1.37 \cdot 10^{11}$	hib	zatvoren
Minerva	2022	8192	$5.4 \cdot 10^{11}$	mon	zatvoren
Megatron-Turing NLG	2022	20480	$5.3 \cdot 10^{11}$	mon	zatvoren
InstructGPT	2022	768-12288	$1.25 \cdot 10^{8}\text{-}1.75 \cdot 10^{11}$	mon	zatvoren
Chinchilla	2022	4096	$7 \cdot 10^{10}$	mon	zatvoren
OPT	2022	768-12288	$1.25 \cdot 10^{8}\text{-}1.75 \cdot 10^{11}$	mon	otvoren

Naziv	God.	Dimenzija VP	Broj parametara	Građa	Otvorenost
UL2	2022	768-4096	$1.2 \cdot 10^{10}$-$2 \cdot 10^{10}$	mon	otvoren
Galactica	2022	2048	$1.2 \cdot 10^{12}$	mon	nedostupan
PaLM	2023	8192	$8 \cdot 10^{9}$-$5.4 \cdot 10^{11}$	mon	zatvoren
GLM-130B	2023	5120	$1.3 \cdot 10^{11}$	mon	otvoren
BLOOM	2023	12288	$1.76 \cdot 10^{11}$	hib	otvoren
LLaMA	2023	4096-8192	$7 \cdot 10^{9}$-$6.5 \cdot 10^{10}$	mod	otvoren
GPT-4	2023	1536-3072	$1.8 \cdot 10^{12}$	mon	zatvoren, API
PaLM 2	2023	12288	$5.4 \cdot 10^{11}$	mod	zatvoren[1]
RWK-World	2023	1024	$1.4 \cdot 10^{9}$	mon	otvoren
LLaMA 2	2023	4096-8192	$7 \cdot 10^{9}$-$7 \cdot 10^{10}$	mod	otvoren
Mistral 7B	2023	768	$7 \cdot 10^{9}$	mon	otvoren
Flan-T5	2024	512-1024	$6 \cdot 10^{7}$-$1.1 \cdot 10^{10}$	mon	otvoren
Flan-PaLM	2024	4096-16384	$8 \cdot 10^{9}$-$5.4 \cdot 10^{11}$	mon	zatvoren
Kosmos-1	2024	4096	$1.6 \cdot 10^{12}$	hib	zatvoren
Dromedary	2024	12288	$1.5 \cdot 10^{12}$	mod	otvoren

Legenda: mon=monolitna; mod=modularna; hib=hibridna

Tablica 2 prikazuje usporedbu nekih od najvažnijih velikih jezičnih modela temeljenih na dubokom učenju, razvijenih od 2018. godine do danas. To je velik, ali moguće i nepotpun popis, s obzirom na to da se novi modeli neprestano razvijaju. Značajke modela poput broja parametara, dimenzija vektorskog prostora i arhitekture mogu utjecati na njihovu učinkovitost u različitim zadacima obrade prirodnog jezika. Također, otvorenost modela ima važnu ulogu u njihovoj dostupnosti i primjeni u istraživačke i komercijalne svrhe. Građa LLM-ova može biti monolitna, modularna ili hibridna, što opisuje kako su komponente modela organizirane i kako međusobno komuniciraju. U monolitnoj arhitekturi sve komponente modela integrirane su u jednu veliku neuronsku mrežu. To je najčešći pristup za LLM-ove jer je jednostavan za implementaciju i može postići dobre performanse. Međutim monolitni modeli mogu biti teški za ažuriranje i održavanje, a mogu biti i manje interpretabilni. U modularnoj arhitekturi model je podijeljen na manje, specijalizirane module. Oni mogu biti trenirani neovisno jedan o drugome i mogu se kombinirati na različite načine za obavljanje različitih zadataka. Modularni modeli su fleksibilniji i interpretabilniji od monolitnih modela, ali se teže

1 Zatvoren, uz mogućnost pristupa modelu i njegovim mogućnostima putem *Google*-ove platforme Vertex AI.

implementiraju i optimiziraju. Hibridni modeli su takve građe koja kombinira elemente monolitne i modularne arhitekture. Naprimjer model može imati monolitnu osnovu s nekoliko modularnih komponenti koje se mogu dodati ili ukloniti po potrebi. Hibridni modeli nude kompromis između fleksibilnosti modularnih modela i performansi monolitnih modela. Trenutno postoji trend k razvoju modularnijih LLM-ova stoga što modularni modeli imaju veću fleksibilnost, interpretabilnost i skalabilnost od monolitnih. Međutim monolitni modeli još su uvijek popularni zbog svoje jednostavnosti i svojih performansi.

3.3.4. Inherentna semantika vektorskog prostora modela DL

Razvoj dubokog učenja i mogućnost prikupljanja proizvoljno velikih tekstnih skupova podataka u proizvoljnim domenama i na mnoštvu jezika, rezultirali su stvaranjem jezičnih modela koji mogu interpretirati i razumjeti jezike. Ključan element tih modela je vektorski prostor koji omogućuje numeričko predstavljanje riječi, fraza, rečenica ili cijelih tekstova (tekstne jedinice), zadržavajući njihove semantičke značajke. U jezičnim modelima koji se temelje na dubokom učenju tekstne jedinice predstavljene su kao vektori u višedimenzionalnom prostoru. Taj pristup omogućuje modelima prepoznavanje sličnosti i odnosa među riječima na temelju njihovih vektorskih reprezentacija. Modeli poput Word2Vec (Mikolov i sur., 2013b), demonstrirali su kako riječi koje se često pojavljuju u sličnim kontekstima mogu biti predstavljene vektorima koji su blizu jedan drugome u vektorskom prostoru. Naprimjer vektori za riječi „kralj” i „kraljica” bit će bliži nego vektori za „kralj” i „stolica” (Mikolov i sur., 2013a). Štoviše, ako vektoru koji odgovara riječi „kraljica” oduzmemo vektor koji odgovara riječi „žena”, dobit ćemo vektor koji je približno jednak vektoru za riječ „kralj”. Dakle, semantika inherentna vektorskim reprezentacijama riječi omogućava da se s vektorskim reprezentacijama radi/računa na način linearne algebre. Ta iznenađujuća pojava najavila je istraživačku eksploziju u području dubokog učenja i jezičnih modela, potpomognutu povećanjem računalne snage toliko potrebne za treniranje neuronskih mreža s mnogo slojeva i dimenzija, odnosno parametara.

Prvi jezični modeli poput Word2Veca pružali su statičke vektorske reprezentacije, no poslije su se pojavili modeli poput BERT-a (engl. *Bidirectional Encoder Representations from Transformers*) koji koriste kontekstne vektorske reprezentacije. Devlin i suradnici (2018) pokazali su da BERT može prilagoditi značenje riječi ovisno o kontekstu u kojem se riječ nalazi. Naprimjer riječ „*bank*” imat će različite vektorske reprezentacije u rečenicama „*I sat*

on the river bank" i „*I need to visit the bank*" (Devlin i sur., 2018).

Usporedba semantičke sličnosti među vektorima ključna je za mnoge primjene u NLP-u. Kosinusna sličnost jedna je od najčešće korištenih mjera za procjenu sličnosti među vektorima. Izračunava se kao kosinus kuta između dva vektora, pri čemu manji kut označava veću sličnost. Ta se mjera široko koristi u npr. pretraživanju informacija, sustavima preporuka i analizama tekstova (P. D. Turney i Pantel, 2010).

Jedan je od izazova u korištenju modela DL njihova interpretabilnost. Modeli često funkcioniraju kao „crne kutije", gdje je teško razumjeti kako dolaze do određenih zaključaka. To je posebno problematično u kritičnim aplikacijama poput medicinske dijagnostike ili pravnog savjetovanja. Neka buduća istraživanja sigurno će se usredotočiti na razvoj modela koji su transparentniji i lakši za interpretaciju (Belinkov i Glass, 2019).

Jezični modeli koji se temelje na dubokom učenju i njihova sposobnost razumijevanja semantičkih odnosa imaju široku primjenu. Tako se u domeni pretraživanja informacija modeli poput BERT-a koriste za poboljšanje relevantnosti rezultata pretrage, u sustavima preporuka vektorske reprezentacije pomažu u identifikaciji sličnih proizvoda ili sadržaja, u analizi sentimenta vektorske reprezentacije riječi omogućuju preciznije razumijevanje tonova i emocija izraženih u tekstu (Devlin i sur., 2018).

Napredak u vektorskim prostorima također se širi na multimodalne modele koji kombiniraju tekstne, slikovne i audiopodatke. Naprimjer model CLIP (engl. *Contrastive Language-Image Pretraining*) kojega je razvio OpenAI koristi zajednički vektorski prostor za predstavljanje tekstova i slika, omogućujući modelu da prepoznaje slike na temelju tekstnih opisa (Radford i sur., 2021). Ti multimodalni pristupi otvaraju nove mogućnosti za integraciju različitih tipova podataka i unaprjeđuju sposobnosti razumijevanja modela DL.

Za široku primjenu modela DL, uz njihovu interpretabilnost, ključna je i njihova robusnost. Modeli moraju biti otporni na promjene u ulaznim podacima i sposobni generalizirati izvan podatkovnoga skupa za treniranje. Taj je izazov posebno izražen u slučajevima kada se modeli primjenjuju na jezike ili domene koji nisu bili uključeni u početno treniranje. Istraživanja u tom području usmjerena su na razvoj tehnika za bolju generalizaciju i prilagodljivost modela (Belinkov i Glass, 2019).

Inherentna semantika vektorskog prostora jezičnih modela koji se temelje na dubokom učenju temelj je za ispunjenje ciljeva ovoga istraživanja.

3.4. Mjere evaluacije

U nastavku su predstavljene mjere koje se standardno koriste za evaluaciju rezultata u području obrade prirodnog jezika. To su mjere koje su relevantne za postupke evaluacije rezultata istraživanja: preciznost, odziv, F1-mjera, matrica zabune, točnost, pogreška, Vennovi dijagrami, Matthewsov koeficijent korelacije.

Preciznost, odziv i F1-mjera glavne su mjere za procjenu učinkovitosti tehnika obrade prirodnoga jezika ili pretraživanja informacija (Hsiao i sur., 2014; Marsi i Krahmer, 2010), ali u upotrebi su i druge mjere. Sve su to sažete statistike, sintetički jednodimenzionalni pokazatelji.

Prepoznavanje semantičke sličnosti uključuje dvije klase dokumenata: one koji su semantički slični s predloškom (dokumentom) i one koji to nisu. Proizvoljna metoda otkrivanja semantičke sličnosti može okarakterizirati dokument kao semantički sličan predlošku („pozitivan“) ili ne („negativan“). To znači da pozitivno označen dokument može biti ili stvarno pozitivan (semantički sličan predlošku) ili lažno pozitivan. Također, negativno označen dokument može biti stvarno ili lažno negativan. Situaciju opisuje dobro poznata matrica zabune (Alpaydin, 2010; Bengio i sur., 2003) za dvije klase dokumenata (tablica 3), no to vrijedi i mnogo općenitije pa će se u nastavku umjesto *dokumenata* koristiti pojam *instance*. Matrica zabune je tablica koja prikazuje performanse klasifikacijskog modela, pokazujući ispravne i pogrešne klasifikacije za svaku klasu.

Tablica 3. Matrica zabune za dvije klase instanci

		Utvrđeno	
		Pozitivno	*Negativno*
Stvarno	*Pozitivno*	*TP*	*FN*
	Negativno	*FP*	*TN*

T=*True*, istinito, F=*False*, lažno

Osnovne evaluacijske mjere performansi takve klasifikacije su (Alpaydin, 2010; Alvarez, 2002; Bašić i Šnajder, 2011; Pejaković, 2009; Sasaki, 2007):

Preciznost (engl. *precision*) je udio točno klasificiranih instanci (*TP*) u skupu pozitivno klasificiranih instanci (*TP*+*FP*). Koristi se i engl. naziv *positive predictive value* (*PPV*)

(Sasaki, 2007).

$$P = PPV = \frac{TP}{TP + FP} \tag{19}$$

Odziv (engl. *recall*) je udio točno klasificiranih instanci u skupu svih pozitivnih instanci (*TP*+*FN*). Drugi nazivi su: osjetljivost (engl. *sensitivity*) (Sasaki, 2007), *hit rate*, *true positive rate* (*TPR*).

$$R = TPR = \frac{TP}{TP + FN} \tag{20}$$

Pored preciznosti i odziva u domeni NLP-a koriste se i druge mjere koje će biti navedene u nastavku. Među njima je najznačajnija F1-mjera.

Točnost (engl. *accuracy*) je udio točno klasificiranih instanci (*TP*+*TN*) u skupu svih instanci (*N*).

$$Acc = \frac{TP + TN}{N} \tag{21}$$

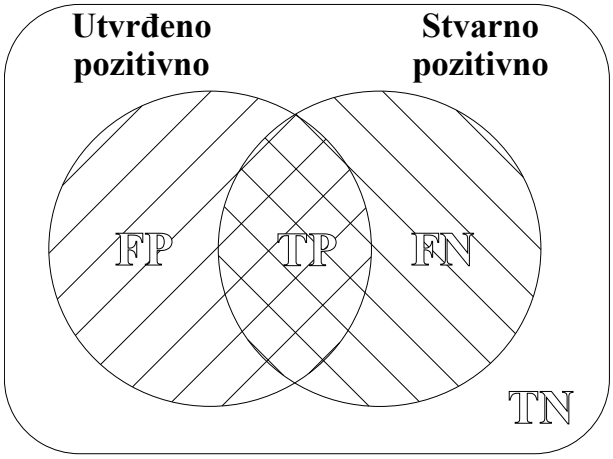
Pogreška (engl. *error*) je udio pogrešno klasificiranih instanci (*FP*+*FN*) u skupu svih instanci.

$$E = \frac{FP + FN}{N} \tag{22}$$

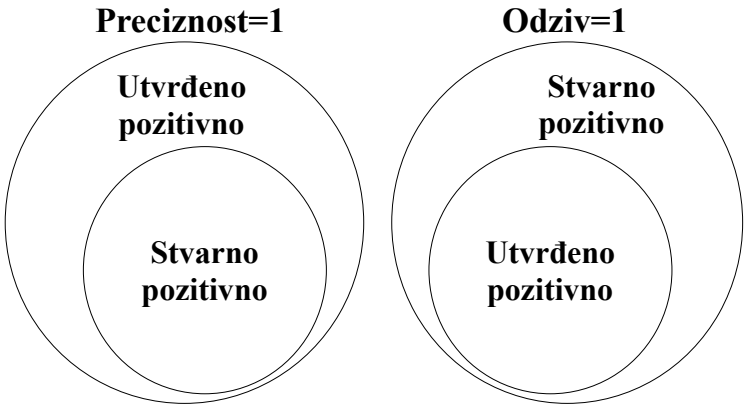
Očito se točnost i pogreška međusobno nadopunjuju u smislu udjela u skupu svih instanci te se znajući jednu veličinu može jednostavno izračunati druga veličina pomoću formule.

$$Acc = 1 - E \tag{23}$$

Potpuniji uvid u performanse sustava (Goutte i Gaussier, 2005) može se dobiti grafičkim prikazom krivulje *preciznost-odziv*. Odnos između preciznosti i odziva može se pregledno vizualizirati (Alpaydin, 2010; Bašić i Šnajder, 2011) Vennovim dijagramima prikazanim na slikama 9 i 10. Dijagramom na slici 9 vizualizirani su odnosi između klasificiranih i stvarnih klasa. U dijagramu na slici 10 lijevi krug vizualizira preciznost kao omjer *TP* i cijeloga kruga (*FP*+*TP*), a odziv vizualizira desni krug kao omjer *TP* i cijeloga kruga (*TP*+*FN*).



Slika 9. Vennov dijagram: *TP*, *TN*, *FP*, *FN*



Slika 10. Vennov dijagram: preciznost i odziv

Specifičnost (engl. *specificity*) je udio točno klasificiranih instanci u skupu svih negativnih instanci. Drugi joj je naziv engl. *true negative rate* (*TNR*).

$$S = TNR = \frac{TN}{TN + FP} \tag{24}$$

F1-mjera ili F_1 je harmonijska sredina preciznosti *P* (engl. *precision*) i odziva *R* (engl. *recall*) (Sasaki, 2007).

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \tag{25}$$

U općem slučaju preciznost i odziv mogu biti od različite važnosti za istraživače. Stoga se navedena formula proširuje faktorom *ß*.

$$F_{ß} = \frac{(ß^2 + 1)\,PR}{ß^2\,P + R} \tag{26}$$

Za $ß$=1 preciznost i odziv imaju jednaku važnost, a gornja formula postaje obična harmonijska sredina F i naziva se F_1. Tipično se, pored F_1, koriste i $F_{0.5}$ (kada se želi naglasiti preciznost: $ß$<1) i F_2 (kada se želi naglasiti odziv: $ß$>1). F_1 je općenito glavna standardna mjera evaluacije kvalitete nekog modela.

Matthewsov koeficijent korelacije (engl. *Matthews Correlation Coefficient*, *MCC*) je nešto je manje poznata, ali ipak standardna mjera za evaluaciju postupaka određivanja sličnosti tekstova i sličnih zadataka, odnosno preciznije, za evaluaciju rezultata binarnih klasifikacija. U tom se smislu može koristiti za evaluaciju postupaka detekcije parafraziranja. U strojnom učenju koristi se kao mjera kvalitete binarnih klasifikacija, a uveo ju je biokemičar Brian W. Matthews (Matthews, 1975). U statistici je ista mjera poznata pod nazivom ***phi* koeficijent** i mjera je povezanosti dviju binarnih[1] varijabli. Izračunava se iz ***hi-kvadrat*[2] vrijednosti** i veličine uzorka. Vrijednosti *phi* koeficijenta kreću se od -1 (savršena negativna korelacija) do 1 (savršena pozitivna korelacija), pri čemu 0 označava da nema povezanosti. *Hi-kvadrat* test govori o tome postoji li povezanost, a *phi* koeficijent koliko je ta povezanost jaka. Formula za izračun *phi* koeficijenta ili *MCC* je sljedeća:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{27}$$

Kako bi se procijenile binarne klasifikacije i njihove matrice zabune, može se upotrijebiti nekoliko statističkih mjera, ovisno o cilju eksperimenata. Točnost i F1 rezultat/mjera izračunati na matricama zabune među najčešće su korištenim metrikama u zadacima binarne klasifikacije. Međutim te statističke mjere mogu opasno pokazivati preoptimistične, napuhane, rezultate, posebno na neuravnoteženim skupovima podataka. Matthewsov koeficijent korelacije (*MCC*) pouzdanija je statistička mjera koja daje visok rezultat samo ako je predviđanje postiglo dobre rezultate u svim četirima kategorijama matrice zabune (istiniti pozitivni, lažni negativni, istiniti negativni i lažni pozitivni), proporcionalno i veličini pozitivnih i negativnih elemenata u skupu podataka. Stoga je *MCC* informativniji i istinitiji rezultat procjene binarnih klasifikacija od točnosti i F1 rezultata (Voxco, 2024).

1 Preciznije: između dviju dihotomnih varijabli (varijabli s dvije kategorije).

2 Hi-kvadrat test (χ^2) je statistički test koji se koristi za ispitivanje povezanosti između dviju kategoričkih varijabli. On pokazuje postoji li statistički značajna povezanost između varijabli, ali ne i koliko je ta povezanost jaka.

4. Istraživački postupak i razvoj metode DLPDM

Ovo poglavlje opisuje tijek istraživačkoga postupka u okviru kojega je razvijena metoda DLPDM (*Deep Learning Paraphrase Detection Method*) koja omogućava detekciju parafraziranja u tekstu. Detaljno su prikazani glavni aspekti istraživanja: (i) razvoj konceptualnog modela metode DLPDM, (ii) tijek istraživačkih eksperimenata i formalizacija postupaka za odabir odgovarajućih parametara i elemenata metode, te (iii) korpusi parafraziranih tekstova koji su neophodni za provođenje eksperimenata. U skladu s time ovo je poglavlje podijeljeno u tri cjeline i tri potpoglavlja. U poglavlju *4.1. Konceptualni model metode DLPDM* opisan je konceptualni model metode i dan je pregled glavnih faza i koraka metode. Potom je u poglavlju *4.2. Razvoj metode DLPDM* pregled tijeka provođenja eksperimenata čija je implementacija detaljno opisana u poglavlju *5. Eksperimenti*. Nadalje, u poglavlju *4.3. Korpusi parafraziranih tekstova* detaljno su opisani korpusi koji su korišteni za evaluaciju postupaka, dok su rezultati evaluacije opisani u poglavlju *6. Rezultati*.

4.1. Konceptualni model metode DLPDM

Glavni je cilj istraživanja definiranje metode koja detektira je li zadani tekst parafraziran i na taj način omogućava otkrivanje plagiranja koje je nastalo parafraziranjem. U ovome potpoglavlju opisan je razvoj konceptualnog modela metode DLPDM.

Metoda DLPDM temelji se na modelima dubokog učenja, a omogućava otkrivanje parafraziranja kroz niz koraka, najprije detektirajući sličnost tekstova na razini dokumenata, a potom detektirajući parafraziranje na razini rečenica. U prvome dijelu istraživačkog postupka razvijen je model metode DLPDM na konceptualnoj razini. Definicija metode na konceptualnoj razini omogućava fleksibilnost u smislu da konceptualni model propisuje korake i parametre metode na metarazini, a potom je u idućoj fazi moguć odabir i uključivanje konkretnih parametara metode. Parametri se odnose na različite tehnike pripreme teksta, različite modele i metode za reprezentaciju teksta, različite mjere sličnosti te različite granične vrijednosti koje se koriste (prag sličnosti). Fleksibilnost konceptualnog modela osigurava modularni dizajn koji omogućuje zamjenu nekoga od parametara metode novim parametrom, dok konceptualni model ostaje nepromijenjen. Primjerice s obzirom na stalni razvoj novih jezičnih modela moguće je korišteni jezični model u budućnosti zamijeniti novim modelom koji može poboljšati točnost detekcije. Tako koncipiran konceptualni model

metode osigurava aktualnost metode i u budućnosti.

Metoda DLPDM propisuje detekciju parafraziranja u tri faze, od kojih svaka propisuje nekoliko koraka. Na konceptualnoj razini definirane su sve faze i koraci, a potom su kroz niz eksperimenata utvrđeni konkretni parametri metode koji daju najbolje rezultate u detekciji parafraziranja.

U **prvoj fazi** provodi se pretprocesiranje dokumenata i priprema teksta za drugu fazu. Polazi se od pretpostavke da postoji dokument koji treba provjeriti, dok se korpus usporedbenih dokumenata može preuzeti iz različitih izvora. U toj, pripremnoj fazi izvodi se ekstrakcija teksta iz dokumenata, učitavaju se datoteke korpusa i neformatirani tekst izdvaja se iz dokumenata korištenjem različitih pristupa prilagođenih ulaznim formatima datoteka. Potom se primjenjuju odgovarajuće metode NLP-a i tehnike pripreme teksta. Neke od standardnih tehnika pripreme teksta u domeni NLP-a su uklanjanje ili zamjena zaustavnih riječi (engl. *stop-words*), lematizacija, uklanjanje riječi od samo jednoga znaka, korjenovanje (engl. *stemming*), ali u ovome istraživanju analizirane su i neke druge tehnike. Kako bi prva faza bila definirana, potrebno je utvrditi koje su metode i tehnike NLP-a najbolje za pripremu teksta u zadatku detekcije parafraziranja i otkrivanja plagijata i na temelju toga odrediti **niz tehnika za obradu i pripremu teksta** $\boldsymbol{O_{txt}=[o_1, ..., o_n]}$. Za definiranje skupa $\boldsymbol{O_{txt}}$ proveden je niz eksperimenata koji su opisani u idućim poglavljima.

U **drugoj fazi** provodi se usporedba cjelovitih dokumenata kako bi se detektirali parovi dokumenata koji su slični. Prvi je korak u toj fazi reprezentirati tekst primjenom odabranoga jezičnog modela, a potom odrediti sličnost primjenom odgovarajuće mjere za mjerenje sličnosti vektorskih reprezentacija teksta. Nadalje, ako izračunata sličnost prelazi prag sličnosti koji je utvrđen putem eksperimenata, smatra se da dokumenti potencijalno sadrže parafrazirane tekstove. Elementi, odnosno parametri koje je potrebno odrediti kako bi svi koraci u drugoj fazi bili definirani jesu **model za vektorsku reprezentaciju teksta,** $\boldsymbol{M^*}$, zatim **mjera sličnosti tekstova (dokumenata)** ***sim***, te **granična vrijednost za prag sličnosti,** $\boldsymbol{\theta^*}$. Za određivanje modela M^* provedeno je opsežno istraživanje i fino podešavanje niza modela temeljenih na dubokom učenju, a također i analiza niza drugih metoda i modela za reprezentaciju teksta. Nadalje, za određivanje najboljih vrijednosti preostalih navedenih parametara provedeni su eksperimenti u kojima su se ispitale razne mjere sličnosti i različite vrijednosti parametara s ciljem identifikacije najboljih parametara.

U **trećoj fazi** detektira se parafraziranje na razini rečenica. U prvom koraku izdvajaju

se rečenice iz odabranih parova dokumenata koji su se pokazali dovoljno sličnima u prethodnoj fazi te se u drugom koraku ispituje detekcija parafraziranja na razini rečenica. Za potrebe otkrivanja parafraziranja na razini rečenica definirana je **nova kompozitna mjera parafraziranja**, nazvana *Deep Learning Composite Paraphrase Measure* (DLCPM). Slično kao i u drugoj fazi, nakon izračunavanja mjere parafraziranja za rečenični par utvrđuje se parafraziranost ovisno o pragu te se, ako izračunata vrijednost mjere DLCPM prelazi određenu vrijednost, rečenice smatraju parafraziranima. Ako je vrijednost mjere DLCPM manja od praga, rečenice se ne smatraju parafraziranima. Pored mjere DLCPM potrebno je za definiranje treće faze odrediti **graničnu vrijednost za prag parafraziranja,** $\boldsymbol{\theta^{**}}$.

Specifičnost predložene metode ogleda se u tome da se utvrđivanje parafraziranja provodi na dvije razine. Najprije se provjerava semantička sličnost na razini cijelih dokumenata. Ako se utvrdi sličnost veća od nekoga prethodno utvrđenog praga, to može ukazivati na potencijalno postojanje parafraziranja u tekstu. Međutim, kako bi se parafraziranje pouzdano detektiralo, nužno je analizirati manje segmente teksta unutar dokumenta. Iz toga razloga uvodi se dodatna provjera koja ispituje sličnost na razini rečenica kao osnovnih misaonih cjelina dokumenta. Temeljna pretpostavka jest da se prikriveno plagiranje radi na manjim segmentima teksta na način da se izmijene pojedine riječi i preuredi rečenična struktura. Stoga je za detekciju parafraziranja na razini rečenica osmišljena, prethodno spomenuta, kompozitna mjera parafraziranja, DLCPM. Sastoji se od više elemenata i omogućuje prepoznavanje obrazaca prikrivenog plagiranja analizom specifičnih izmjena u tekstu. Za razliku od metode, koja je sveobuhvatni pristup usporedbi tekstova, mjera DLCPM je specifičan alat koji utvrđuje postojanje parafraziranja između dvaju kraćih tekstova (rečenica ili paragrafa) kombiniranjem više aspekata sličnosti.

DLCPM mjera integrira tri aspekta, odnosno komponente u okviru kojih je moguće detektirati parafraziranje: (i) sličnost vektorskih reprezentacija teksta dobivenih iz jezičnog modela dubokog učenja, izračunatu preko kosinusne mjere sličnosti vektorskih reprezentacija, koja detektira **semantičku sličnost** na temelju kontekstnog značenja; (ii) **sličnost tekstova** temeljenu na algoritmu *Greedy Word Tiling*, koji je prilagođen za rad s riječima kao temeljnim jedinicama umjesto nizova znakova (koji se koriste kod standardnog algoritma sličnosti tekstova *Greedy String Tiling*), koja detektira strukturnu i leksičku sličnost; te (iii) **sličnost dužine teksta** mjerena relativnim brojem istih riječi, koja pruža dodatnu informaciju o fizičkoj sličnosti tekstova. Te tri komponente spajaju se u ukupnu mjeru parafraziranja putem

ponderirane kvadratne sredine, gdje je sličnost vektora pozitivno ponderirana kako bi se naglasila semantička važnost, sličnost po *Greedy Word Tiling* negativno ponderirana kvadriranjem kako bi se smanjio njezin utjecaj u slučajevima manje relevantnih podudarnosti, dok se sličnost dužine ostavlja bez dodatnog ponderiranja. Taj pristup osigurava da mjera za ulazne vrijednosti iz intervala [0, 1] daje rezultate manje ili jednake aritmetičkoj sredini, ali istodobno naglašava utjecaj većih vrijednosti, čime se postiže ravnoteža između različitih aspekata sličnosti. Konkretna formula mjere DLCPM iskazana je u jednadžbi (43) u potpoglavlju *4.2.3. Istraživanje postupaka detekcije parafraziranja na razini rečenica*. Primjena mjere DLCPM unutar metode DLPDM omogućuje sveobuhvatnu analizu sličnosti koja nadilazi ograničenja pojedinačnih mjera sličnosti.

Nakon definiranja konceptualnog modela metode DLPDM i mjere parafraziranja DLCPM provodi se u drugome dijelu istraživačkog postupka niz eksperimenata kroz koje je utvrđeno koji konkretni postupci obrade teksta, jezični modeli, mjere sličnosti te pragovi sličnosti daju najbolje rezultate u otkrivanju parafraziranja te je na kraju definirana i konkretna metoda DLPDM s točno propisanim elementima. Tijek istraživanja i formalizacija određenih koraka opisana je u nastavku.

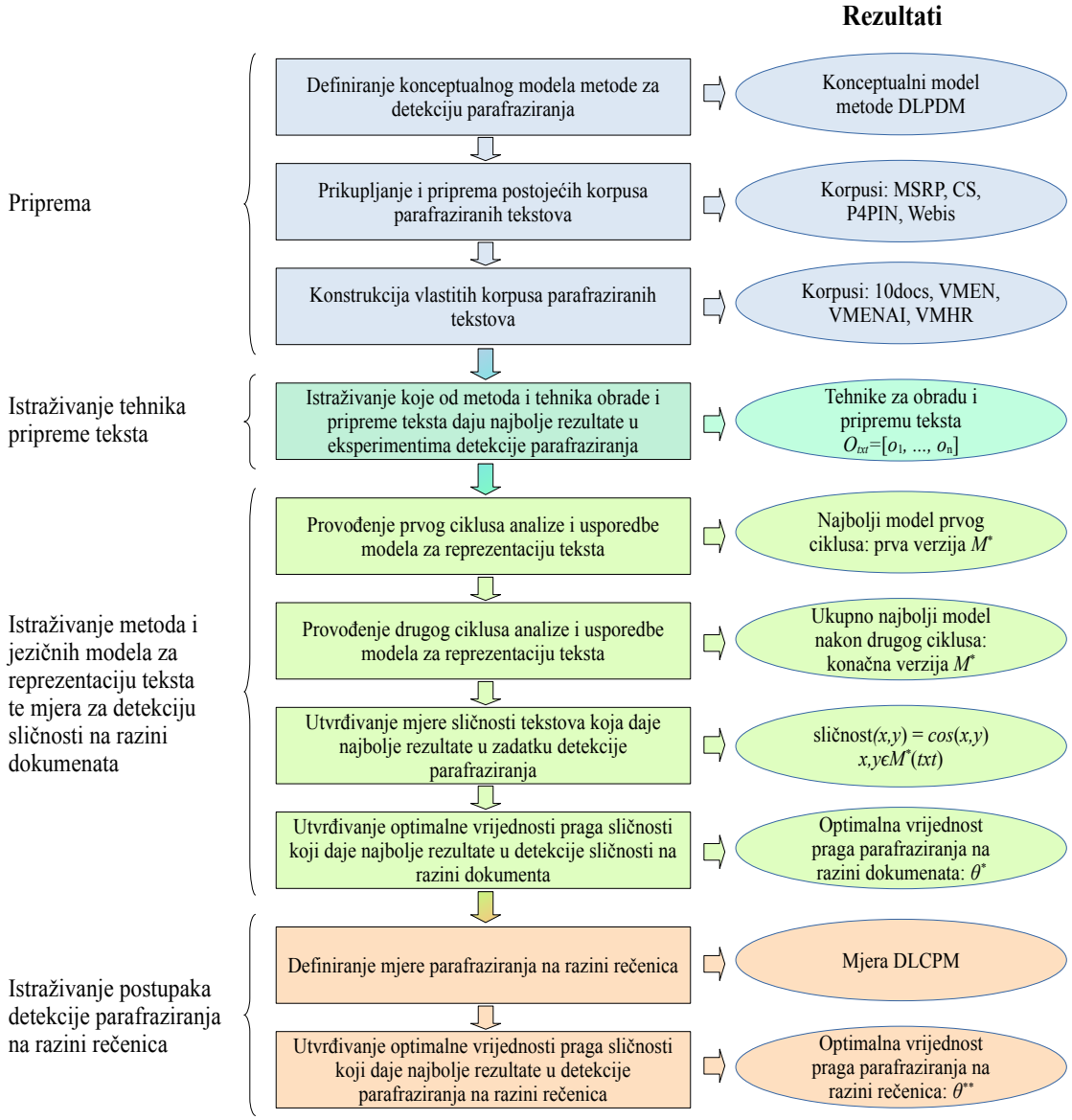
4.2. Razvoj metode DLPDM

U ovom potpoglavlju opisani su istraživački postupci u okviru kojih se razvijala metoda DLPDM. Istraživanje je provedeno u nekoliko etapa s ciljem formalizacije svih koraka metode DLPDM i određivanja parametara metode. Etape istraživanja prate faze predložene metode (priprema teksta, detekcija sličnosti na razini dokumenata i detekcija parafraziranja razini rečenica).

Dakle, metoda DLPDM razvijana je u okviru istraživanja koje se odvijalo u tri etape: (i) istraživanje metoda i tehnika pripreme teksta za učinkovitu detekciju parafraziranja, (ii) istraživanje i treniranje jezičnih modela i metoda za reprezentaciju teksta te analiza mjera sličnosti tekstova za detekciju sličnosti na razini dokumenta, (iii) istraživanje postupka mjerenja parafraziranja tekstova na razini rečenica. Za svaku od etapa proveden je niz eksperimenata čija su implementacija i izvođenje detaljno opisani u poglavlju *5. Eksperimenti*.

Tijek istraživačkoga postupka može se slikovito opisati i dijagramom toka prikazanim

na slici 11. *Vizualizacija tijeka istraživačkog postupka*. Dijagram toka prikazuje proces u tri etape za semantičku usporedbu dokumenata i rečenica unutar korpusa.



Slika 11. Vizualizacija tijeka istraživačkog postupka

4.2.1. Istraživanje metoda i tehnika pripreme teksta za detekciju parafraziranja

Pretpostavimo dakle da postoji korpus[1] dokumenata s kojim se uspoređuje dokument

1 Dohvaćanje relevantnoga korpusa nije obuhvaćeno ovim istraživanjem, koje bi time postalo suviše široko. Koncepcijski, relevantni korpus s kojim se dokument uspoređuje moguće je formirati pristupom jednoj ili više relevantnih baza putem API poziva, korištenjem interne baze radova (fakulteta, sveučilišta, nacionalnog

koji je predmet provjere kako bi se utvrdilo je li došlo do plagiranja parafraziranjem. Neka je $C=\{D_1, D_2, ..., D_n\}$ korpus dokumenata, gdje svaki D_i predstavlja pojedinačni dokument u skupu $D_i \in C$. Dokument D koji nije element skupa C, $D \notin C$ i koji je predmet provjere, uspoređuje se s elementima iz korpusa C kako bi se utvrdilo je li došlo do plagiranja putem parafraziranja. Nakon što programski sustav za otkrivanje plagiranja zaprimi dokument, on prolazi kroz pripremnu obradu kojom se dolazi teksta pogodnog za eksperimente.

Provjeravani dokument D i dokumenti s kojima se on uspoređuje $D_i \in C$ pretvaraju se u neformatirani tekst postojećim besplatnim alatima otvorenog koda: *antiword* za format „doc“; *docx2txt* za format „docx“, *soffice* za format „rtf“, *pdftotxt* za format „pdf“, *odt2txt* za format „odt“ te *html2text* za format „html“ i *BeautifulSoup* za format „json“ ulaznog dokumenta. Neka D^* predstavlja ulazni dokument iz unije skupova dokumenata $D^* \in C \cup D$ u nekom od podržanih formata (*txt*, *rtf*, *doc*, *docx*, *pdf*, *odt*, *html*, *json*). Dokument D^* se pretvara u neformatirani tekst koristeći odgovarajući alat otvorenog koda, ovisno o formatu: $T=f(D^*)$ gdje je T neformatirani tekst dobiven iz dokumenta D^*, a $f(D^*)$ funkcija koja predstavlja odgovarajući alat za pretvorbu, definirana na sljedeći način:

$$f(D^*)=\begin{cases} antiword(D^*) & \text{ako je D}^* \text{ u formatu doc} \\ docx2txt(D^*) & \text{ako je D}^* \text{ u formatu docx} \\ soffice(D^*) & \text{ako je D}^* \text{ u formatu rtf} \\ pdftotxt(D^*) & \text{ako je D}^* \text{ u formatu pdf} \\ ocrmypdf(D^*) & \text{ako je D}^* \text{ u slikovnom formatu pdf} \\ odt2txt(D^*) & \text{ako je D}^* \text{ u formatu odt} \\ html2text(D^*) & \text{ako je D}^* \text{ u formatu html} \\ BeautifulSoup(D^*) & \text{ako je D}^* \text{ u formatu json} \\ D^* & \text{ako je D}^* \text{ već u formatu txt} \end{cases} \quad (28)$$

U slučaju da je dokument D^* u formatu „pdf” nečitljiv, primjerice zbog toga što je sastavljen od slikovnog oblika teksta, dokument D^* rastavlja se na pojedine stranice (slike) $D^*=[S_1, S_2, ..., S_m]$, gdje S_j označava pojedinu stranicu u obliku slike, nad kojima se provodi optičko prepoznavanje teksta (engl. *optical character recognition*, OCR) kako bi se dobio neformatirani tekst T, tj. $T_j=OCR(S_i)$ za svaku stranicu S_j, gdje T_j predstavlja tekst dobiven iz slike S_j. Konačni neformatirani tekst T takva pdf dokumenta D^* dobije se spajanjem svih tekstova T_j nakon OCR procesa. U konačnici se neformatirani tekst pretvara u mala slova

repozitorija) te web-upitima u opće i specijalizirane pretraživače poput Google Scholara, Semantic Scholara, Science Direct, IEEE Explore Digital Library, Deep Dyve, ArXiv, Scopus, Web of Science i sl.

T'=lower(*T*).

Dodatno je u okviru istraživanja provedena analiza i usporedba različitih tehnika NLP-a za pretprocesiranje tekstova, kao što su korištenje n-grama riječi, uklanjanje ili zamjena brojeva, korištenje samo nekih tipova riječi, uklanjanje ili zamjena zaustavnih riječi (*stop-words*), lematizacija, uklanjanje riječi od samo jednoga znaka, korištenje morfoloških transformacija WordNeta, korjenovanje (engl. *stemming*), korištenje hiperonima i homonima. Skup mogućih metoda i tehnika obrade i pripreme teksta zapravo je vrlo širok, a za određivanje najboljih tehnika proveden je skup eksperimenata. Detalji eksperimenata opisani su u poglavlju *5. Eksperimenti* te je utvrđeno koje od tehnika pripreme tekstova pridonose učinkovitoj detekciji parafraziranja. Konačan rezultat prve etape istraživanja je niz metoda i tehnika za obradu i pripremu teksta, $\boldsymbol{O_{txt}=[o_1, ..., o_n]}$ koji definira niz tehnika i njihov redoslijed u obradi teksta.

4.2.2. Istraživanje jezičnih modela za reprezentaciju teksta i mjera za detekciju sličnosti na razini dokumenata

Za potrebe definiranja i implementacije druge faze metode DLCPM potrebno je istražiti koji model za reprezentaciju teksta daje najbolje rezultate u detekciji sličnosti i parafraziranja. U tu svrhu provedeno je istraživanje u **dva ciklusa**. U **prvome ciklusu testirano je 60 različitih modela** u zadatku detekcije parafraziranja. U testiranje su uključeni različiti modeli i metode navedeni u skupovima μ_1 i μ_2.

Neka je μ_1 skup modela[1] koji koriste vektorsku reprezentaciju dokumenata i jezične modele temeljene na dubokom učenju, μ_1={cosbert, edbert, mdbert, cosuse, eduse, mduse, wmdft, wmdd2vwdbow, wmdd2vwdm, wmdglw, coselmo, mdelmo, edelmo, mdlaser, wmdw2vcbow, edlaser, coslaser, wmdw2vsg, scsd2vwdbow, scsd2vwdm, scsft, scsglw, scsw2vcbow, scsw2vsg, cosgld, edgld, edd2vwdbow, cosd2vwdbow, mdd2vwdbow, mdgld, cosd2vwdm, mdd2vwdm, cosglw, cosw2vcbow, edglw, mdglw, edw2vcbow, mdw2vcbow, cosft, edft, mdft, edd2vwdm, cosw2vsg, mdw2vsg, edw2vsg, mdd2vddm, mdd2vddbow, cosd2vddm, edd2vddm, cosd2vddbow, edd2vddbow}. Neka je μ_2 skup drugih pristupa, odnosno metoda koje koriste drukčije načine reprezentacije teksta, a to su μ_2={le, tfidf, lsi, rp, jacc, lev, gwt, hdp, lda}. Veći dio modela iz skupa μ_1 temelji se na transformerima, dok su elementi skupa μ_2 neke od najpoznatijih tradicionalnih metoda za reprezentaciju teksta. Skup

1 Vidi Dodatak: Kratice.

μ_2 uveden je u istraživanje ponajprije kako bi se mogli usporediti rezultati modela temeljenih na dubokom učenju i transformerima s klasičnim pristupima reprezentacije teksta.

Nadalje, za model za koji su se pokazali najbolji rezultati proveden je **drugi ciklus eksperimenata, u okviru kojega su istražene mogućnosti 145 inačica toga najboljeg modela** kako bi se utvrdio najbolji model za reprezentaciju teksta u zadacima detekcije parafraziranja. Opis provedenih eksperimenata dan je u poglavlju *5. Eksperimenti*, a na temelju rezultata prikazanih u poglavlju *6. Rezultati* odabran je najbolji model, $\boldsymbol{M}^*$.

Nakon toga dodatno su istražene mogućnosti različitih mjera sličnosti tekstova. Pri tome je cilj ujedno bio i identificirati mjeru sličnosti koja daje najbolje rezultate za zadatak detekcije sličnosti na razini dokumenta. Dakle nakon što se dva teksta nekim modelom upare sa svojom vektorskom reprezentacijom te se time steknu uvjeti za izračun sličnosti tih dvaju tekstova, sličnost se može računati na mnoštvo načina, tj. koristeći razne metrike udaljenosti i sličnosti. U slučaju korištenja metrika udaljenosti, potreban je još jedan korak kojim se iz rezultata udaljenosti računski dolazi do rezultata sličnosti, za što također postoji više načina. Postavljaju se dva istraživačka pitanja.

- Utječe li (i u kojoj mjeri) odabir mjere udaljenosti/sličnosti na uspješnost otkrivanja sličnih tekstova?
- Utječe li vrsta transformacije između rezultata udaljenosti u rezultate sličnosti na uspješnost otkrivanja sličnih tekstova?

U provedenim eksperimentima različite su mjere sličnosti (udaljenosti) s istim vektorskim reprezentacijama davale različite rezultate. Stoga je trebalo dodatnim eksperimentima dokazati da je neka od mjera sličnosti (udaljenosti) ili bolja od drugih ili je odabir mjere sličnosti (udaljenosti) irelevantan u smislu dobivenih rezultata. Potvrda da rezultati većinom ne ovise o korištenoj mjeri sličnosti (udaljenosti) dokazana je eksperimentalnim putem i prezentirana u radovima (Vrbanec i Meštrović, 2021a, 2023).

Kao mjere sličnosti u ovome istraživanju istražene su kosinusna i meka kosinusna sličnost, a kao mjere udaljenosti *Manhattan* (Krause, 1986), Euklidska (Huang, 2008) i *Word Mover's Distance* (WMD) (Kusner i sur., 2015).

Budući da su u istraživanju korištene i mjere sličnosti i mjere udaljenosti, kod mjera udaljenosti potrebna pretvorba rezultata udaljenosti u rezultate sličnosti izvršena je najjednostavnije, komplementarno, kao $S=1-D$ (29), s obzirom na to da su $S,D \in [0,1]$. Isprobani su i drugi mogući načini pretvorbe (bazni: $S=1/(1+D)$ (30) i kutni: $S=2\arccos D/\pi$

(31)), ali su dali iste rezultate, pa je odabrana najmanje računalno zahtjevna pretvorba.

U nastavku su navedene mjere sličnosti, odnosno udaljenosti te način na koji su one kombinirane u nizu eksperimenata.

- za metode iz μ_1:

$$\mathrm{sim}(D, D_i) = \begin{cases} \cos(v_D, v_{D_i}) & \text{za kosinusnu sličnost} \\ 1 - Euclidean(v_D, v_{D_i}) & \text{za Euklidsku sličnost} \\ 1 - Manhattan(v_D, v_{D_i}) & \text{za Manhattan sličnost} \\ Soft\,Cosine(v_D, v_{D_i}) & \text{za Soft Cosine sličnost} \\ 1 - WMD(v_D, v_{D_i}) & \text{za Word Mover's Distance} \end{cases} \tag{32}$$

- za metode iz μ_2:

$$sim(D, D_i) = \mu(D, D_i) \tag{33}$$

Dio postupka provjere napisan pseudokodom je sljedeći:

```
Ako je metoda iz μ1:
    Izračunaj kosinusnu sličnost, meku kosinusnu sličnost.
    Izračunaj Euklidsku udaljenost, Manhattan udaljenost, WMD udaljenost.
    Za svaki rezultat udaljenosti:
        Izračunaj sličnost.
Ako je metoda iz μ2:
    Koristi metodu za izračun sličnosti (npr. Jaccardova sličnost).
```

Svaka metoda kao samostalna mjera sličnosti ili par koji se sastoji od dviju komponenata – model i mjera sličnosti/udaljenosti za svaki korpus, daje određeni numerički (preciznije decimalni) rezultat iz segmenta [-1, 1]. Taj se rezultat pretvara u binarni oblik, odnosno element iz skupa {0, 1}, koji klasificira rezultat kao sličan ili ne. Binarizacija se provodi na temelju praga sličnosti θ čija je vrijednost usuglašena s karakteristikama odabranoga jezičnog modela, određujući granicu između semantički sličnih i nesličnih tekstova na temelju inherentnih svojstava modela. Taj pristup osigurava da metoda DLPDM sistematski pretvara kontinuirane mjere sličnosti u binarne oznake, čime se omogućuje efikasna detekcija sličnosti i parafraziranja. Za razliku od nekih programskih sustava, poput *Turnitina*, koji granične vrijednosti često definiraju subjektivno na temelju iskustva istraživača (Turnitin AI Technical Staff, 2023), metoda DLPDM koristi prag koji se temelji na karakteristikama jezičnoga modela, čime se postiže dosljednost i ponovljivost rezultata analize sličnosti tekstova.

Kako u konačnici postupak treba detektirati je li tekst sličan ili ne, potrebno je binarizirati rezultat koji se dobije ne temelju mjerenja sličnosti. Semantička sličnost $sim(D, D_i)$ pretvara se u binarnu vrijednost $b(D, D_i)$ na temelju granične vrijednosti θ. Eksperimenti za taj segment konstruirani su tako da se određuje granična vrijednost θ na način da se u petlji varira od 0 do 1 s korakom 0.01 kako bi se definirala granica između sličnih i nesličnih dokumenata. Za svaku graničnu vrijednost θ vrijedi

$$b(D, D_i) = \begin{cases} 1 & \text{ako } \text{sim}(D, D_i) > \theta \\ 0 & \text{ako } \text{sim}(D, D_i) \leq \theta \end{cases} \tag{34}$$

Dio postupka koji se odnosi na binarizaciju rezultata napisan pseudokodom je sljedeći:

```
Za svaku metodu μ:
    Variraj graničnu vrijednost θ od 0 do 1 s korakom 0.01.
    Ako je sličnost veća od θ:
        Postavi binarnu vrijednost na 1. (parafrazirano).
    Inače:
        Postavi binarnu vrijednost na 0. (nije parafrazirano).
```

Na temelju eksperimenata i rezultata prikazanih u poglavljima *5. Eksperimenti* i *6. Rezultati* detektirat će se najbolja mjera sličnosti s oznakom ***sim*** i vrijednost granične vrijednosti $\boldsymbol{\theta}$.

4.2.3. Istraživanje postupaka detekcije parafraziranja na razini rečenica

U zadnjoj etapi istraživanja istraženi su postupci za detekciju parafraziranja na razini rečenica. U okviru tog postupka definirana je kompozitna mjera sličnosti, DLCPM, koja je opisana u potpoglavlju *4.1. Konceptualni model metode DLPDM*. U ovome poglavlju detaljno su razrađene i opisane komponente koje ulaze u izračun parafraziranja na razini rečenica.

Neka su $S_D=[s_1, s_2, ..., s_p]$ i $S_{Di}=[s_1', s_2', s_q']$ liste rečenica dokumenata D i D_i čije su sličnosti preko granične vrijednosti θ^* u smislu parafraziranosti utvrđene u drugoj fazi.

i) Izračunavanje semantičke sličnosti tekstova

Za izračunavanje semantičke sličnosti tekstova koristi se vektorska reprezentacija rečenica primjenom odabranog modela M^* i mjera sličnosti koja je u ovome slučaju odmah postavljena kao kosinusna mjera. Dakle, formula uključuje korištenje modela M^*, tj. vektorskih reprezentacija rečenica sj, s'_k i kosinusne sličnosti.

$$sim_{M^*}(s_j, s'_k) = cos(M^*(s_j), M^*(s'_k)) \quad (35)$$

Dio postupka koji se odnosi na računanje semantičke sličnosti tekstova napisan pseudokodom je sljedeći:

```
Koristi najbolji model M* i izračunaj kosinusnu sličnost između vektorskih
reprezentacija rečenica.
```

ii) Izračunavanje sličnosti na temelju mjere GWT mjereno riječima

Sličnost na temelju mjere GWT prepoznaje dijelove rečenica koji se preklapaju. Detaljni postupak izračuna opisan je u nastavku.

Neka su s_j i s'_k dvije rečenice iz lista S_D i S_{Di}.

Definicija tokena

Definiramo funkciju r(s) koja koristi regularni izraz *re.findall(r'\w+', s)* za ekstrakciju riječi iz rečenice *s*, stvarajući niz tokena (riječi):

$$T_j = r(s_j) = re.findall(r'\w+', s_j) \quad (36)$$

$$T_k = r(s'_k) = re.findall(r'\w+', s'_k) \quad (37)$$

Tako dobivamo nizove tokena $T_j=[t_{j1}, t_{j2}, ..., t_{jm}]$ i $T_k=[t_{k1}, t_{k2}, ..., t_{kn}]$ gdje svaki token t_{ji} i t_{ki} predstavljaju pojedinačnu riječ[1] u rečenicama s_j i s'_k.

Inicijalizacija tokenskih nizova i oznaka te minimalnog praga podudaranja

Svaki token t_{ji} iz T_j i t_{ki} iz T_k treba biti označen kako bi algoritam mogao pratiti koje su riječi već bile uparene u procesu podudaranja. Svakom se tokenu pridružuje oznaka *matched* koja označava je li riječ već bila uparena, uz inicijalnu vrijednost *False,* tj. *mached = False,*

$$\forall t_{ji} \in T_j : \text{matched}(t_{ji}) = False \quad (38)$$

$$\forall t_{ki} \in T_k : \text{matched}(t_{ki}) = False \quad (39)$$

što omogućava algoritmu praćenje podudarnosti riječi tijekom izvođenja GWT algoritma.

Definira se *početni minimalni prag za podudaranje*[2] kao *max_match*=1.

---

1 Token je, u pravilu, riječ, no može biti i nešto što nije riječ, tj. što regularni izraz „prepozna" kao riječ. Iznimno je malo tokena koji u stvarnosti nisu riječ. U stručnoj literaturi takvi se tokeni nazivaju opojavnici.

2 Minimalni prag za podudaranje odnosi se na najmanji broj uzastopnih tokena (riječi, opojavnika) koji moraju biti podudarni da bi se ta sekvenca smatrala značajnim podudaranjem. Naprimjer, ako je minimalni prag podudaranja postavljen na 3, to znači da moraju postojati najmanje tri uzastopna podudarna tokena da bi se sekvenca uparila. Taj prag podudaranja postavljen je na početku algoritma i prilagođava se tijekom rada algoritma.

Glavna petlja (traženje podudaranja)

Dok postoji podudaranje:

prolazi se kroz sve neuspoređene tokene iz T_j i T_k.

Ako su riječi identične, računa se dužina slijeda podudarnih riječi sim_{result} i suma svih podudaranja com_{result}.

$$sim_{result} = sim_{result} + com_{result} \tag{40}$$

gdje sim_{result} broji dužinu trenutne sekvence uzastopnih podudaranja tokena između T_j i T_k, a com_{result} je ukupni zbroj sličnosti za sve sekvence podudaranja koje su pronađene.

Ako je slijed podudarnih riječi duži od trenutnoga maksimalnog podudaranja *max_match*, ažurira se maksimalno podudaranje i lista podudaranja.

Nakon što su pronađena sva podudaranja, označavaju se riječi kao uspoređene.

Računanje rezultata GWT

Računa se ukupna suma podudaranja svih uparenih riječi T_j i T_k.

$$\mathrm{sim}_{GWT}(T_j, T_k) = \sum (com_{result}) \tag{41}$$

Dio postupka koji se odnosi na računanje sličnosti na temelju mjere GWT mjereno riječima napisan pseudokodom je sljedeći:

```
Za svaku rečenicu s_j u dokumentu D i svaku rečenicu s'_k u dokumentu D_i:
    Pretvori rečenice u tokene.
    Inicijaliziraj oznake podudaranja za svaki token.
    Dok god postoji podudaranje između tokena:
        Pronađi niz podudarnih riječi.
        Ažuriraj ukupnu sličnost.
    Izračunaj GWT sličnost za podudarnost tokena.
```

iii) Izračunavanje sličnosti na temelju dužine rečenica

Treća komponenta u mjeri odnosi se na sličnost na temelju dužine rečenice mjereno riječima. U nastavku je dana formula za izračun treće komponente mjere sličnosti.

Sličnost dužine dviju rečenica $sim_{lenght}(s_j, s_k')$ računa se kao

$$\mathrm{sim}_{lenght} = 1 - \frac{2 \cdot \left| \left(L(s_j) - L(s'_k) \right) \right|}{L(s_j) + L(s'_k)} \tag{42}$$

gdje su $L(s_j)$ i $L(s'_k)$ brojevi riječi u rečenicama s_j i s'_k, a s_j i s'_k rečenice iz lista S_D i S_{Di}.

Dio postupka koji se odnosi na računanje sličnosti na temelju dužine rečenice napisan pseudokodom je sljedeći:

```
Za svaku rečenicu sj u dokumentu D i svaku rečenicu s'k u dokumentu Di:
    Izračunaj sličnost dužine kao:
        1 - (2*|dužina(sj)-dužina(s'k)|) / (dužina(sj)+dužina(s'k))
```

iv) Definiranje mjere DLCPM

Na temelju prethodne tri komponente za određivanje različitih aspekata sličnosti tekstova, definira se kompozitna mjera parafraziranja DLCPM. Za dvije rečenice ili dva znakovna niza s_j i s'_k mjera se definira kombinacijom tri vrste sličnosti na sljedeći način:

$$dlcpm(s_j, s'_k) = \sqrt{\frac{2 \cdot (\text{sim}_{M^*}(s_j, s'_k))^2 + (\text{sim}_{GWT}(s_j, s'_k))^4 + (\text{sim}_{length}(s_j, s'_k))^2}{4}} \tag{43}$$

Mjera je definirana na način da komponenta semantičke sličnosti sim_{M^*} dvostruko utječe na ukupni rezultat (pozitivno ponderiranje), GWT mjera sličnosti sim_{GWT} ima umanjeni utjecaj (negativno ponderiranje) pomoću kvadriranja, dok mjera sličnosti dužine rečenica sim_{lenght} nije ponderirana.

Dio postupka koji se odnosi na računanje parafraziranja na temelju mjere DLCPM napisan pseudokodom je sljedeći:

```
Ukupna sličnost je ponderirana kvadratna sredina triju mjera sličnosti:
    dlcpm = sqrt(2(simM*)^2+(simGWT)^4+(simlength)^2)
```

v) Binarizacija rezultata mjere parafraziranja za rečenice

Nadalje, za svaki par rečenica (s_j, s'_k), računa se binarna vrijednost $b(s_j, s'_k)$ na način da ako je $dlcpm(s_j, s'_k) > \theta^{**}$, gdje je θ^{**} granična vrijednost koja se izračunava putem niza eksperimenata slično kao i vrijednost θ^* iz prethodne faze. Konačno, provjeravana rečenica iz rečeničnog para (provjeravana rečenica, potencijalno izvorna rečenica) smatra se parafraziranom (vrijednost 1); u suprotnom se smatra neparafraziranom (vrijednost 0).

$$b(s_j, s'_k) = \begin{cases} 1 & \text{ako} \quad \text{sim}_{total}(s_j, s'_k) > \theta^{**} \\ 0 & \text{ako} \quad \text{sim}_{total}(s_j, s'_k) \leq \theta^{**} \end{cases} \tag{44}$$

Dio postupka koji se odnosi na binarizaciju rezultata mjere DLCPM napisan pseudokodom je sljedeći:

```
Ako je ukupna sličnost veća od praga θ**:
    Postavi binarnu vrijednost na 1 (rečenica je parafrazirana).
Inače:
    Postavi binarnu vrijednost na 0 (rečenica nije parafrazirana).
```

Nakon što je implementirana mjera DLCPM i svi navedeni postupci, na temelju provedenih eksperimenata određen je prag parafraziranja θ^{**}. Time je definirana i treća faza metode DLPDM.

Na temelju opisanih etapa istraživanja za sve tri faze metode DLPDM provedeni su eksperimenti koji su opisani u poglavlju *5. Eksperimenti*. Za provođenje eksperimenata potrebno je imati korpuse parafraziranih tekstova. Korpusi koji su korišteni u ovome istraživanju prezentirani su u sljedećem poglavlju.

4.3. Korpusi parafraziranih tekstova

Za eksperimente svih vrsta korištena su četiri javno dostupna korpusa parafraziranih tekstova te dva nova korpusa modelirana za ovo istraživanje (Vrbanec i Meštrović, 2021a, 2023). Time su obuhvaćeni svi javno dostupni korpusi i podatkovni skupovi koji su pripremljeni za zadatak utvrđivanja parafraziranja: *Clough & Stevenson* (Clough i Stevenson, 2009), *Microsoft Research Paraphrase Corpus* (Dolan i Brockett, 2005), *Webis-11* (Burrows i sur., 2013) i *Paraphrase for Plagiarism Including Negatives examples* (Sánchez-Vega i sur., 2019). Budući da ti korpusi imaju vrlo različita obilježja, a rezultati nisu bili konzistentni[1], nametnula se potreba za dodatnim pouzdanim korpusom kome se može vjerovati. Stoga je u sklopu istraživanja načinjen novi korpus od 200 dokumenata pod nazivom VMEN korpus koji je u završnoj fazi istraživanja modificiran u VMENAIA[2]. Na početku istraživanja konstruiran je još jedan razvojni korpus, nazvan *10docs* koji se sastoji od deset dokumenata, preciznije tekstovi vijesti s *online*-portala (Vrbanec & Meštrović, 2021a), a koji je korišten samo za određivanje i podešavanje hiperparametara i ubrzanje programiranja. Oba nova korpusa javno su dostupna na URL https://vrbanec.com/corpora/.

1 Uspoređujući rezultate parova tekstova čiji se rezultati eksperimenata nisu poklapali sa službenim oznakama, kod nekih korpusa poput MSRP, a posebno kod Webisa, službene oznake previše često nisu odgovarale ljudskom sudu.

2 Detaljnije u poglavlju *4.3.5. Korpusi VMEN i VMENAIA*.

4.3.1. Razvojni korpus 10docs

Za potrebe ovoga istraživanja, ponajprije za razvoj metode DLPDM i programskog, sustava koji je podržava, nužno je imati radni, razvojni korpus tekstova manjih dimenzija radi brzoga prototipiranja, pri čemu su neki tekstovi sličnoga sadržaja dok su drugi sasvim drugačijega, a kako bi se na jednostavan i neupitan način vidjelo ponašaju li se dobiveni rezultati u skladu sa zdravom logikom, tj. grupiraju li se oko istih tema, a razilaze oko različitih. Tako je s različitih novinskih portala dohvaćeno deset različitih tekstova (vijesti) koji sadrže nekoliko međusobno povezanih tema i jednu vijest koja je svojevrsni uljez u tematskom smislu (koji je sadržajno bitno različit od svih ostalih). Sličnost tekstova na semantičkoj razini, o čemu odlučuje ljudski um, trebala bi se reflektirati i na numeričke vrijednosti sličnosti izračunate pomoću metoda i mjera koje su se u trenutku stvaranja korpusa tek trebale razviti. Ovaj korpus nazvan *10docs* služio je u svrhu prototipiranja programa, testiranja smislenosti i prikladnosti postupka koji se razvijao, točnosti izračuna mjera sličnosti i udaljenosti, testiranja stvaranja i ispravnosti reprezentacije riječi u obliku vektorskih reprezentacija modela dubokog učenja stvorenih iz korpusa tekstova, te za određivanje najboljih hiperparametara programskog sustava stvaranoga i korištenoga za brojne eksperimente.

Tablica 4. Primjeri rečenica iz korpusa 10docs

Rečenica	Izvor	Tema
Trump withdraws from Trans-Pacific Partnership amid flurry of orders.	1./*t02.txt*	Trump
Trump signs order withdrawing from TPP, reinstate 'Mexico City policy' on abortion.	1./*t03.txt*	Trump
"Lori Wallach, director of Public Citizen's Global Trade Watch, said it would bury the moldering corpse" of the Pacific deal, though she expressed concern about how Nafta would be renegotiated.	24./*t04.txt*	Trump
We just had probably the most incredible meeting of our careers," Sean McGarvey, president of North America's Building Trades Unions, said.	26./*t04.txt*	Trump
Theresa May will bring Brexit forward by two weeks in change to Article 50 timetable.	1./*t01.txt*	Brexit
LONDON — Easily winning a crucial vote among lawmakers, Prime Minister Theresa May was well on her way Wednesday to winning the parliamentary approval that Britain's highest court said she needed before she could begin talks on ending more than four decades of European integration.	2./*t08.txt*	Brexit
Theresa May, the prime minister, has done little to prepare voters for this debate.	12./*t07.txt*	Brexit
Parliament will rightly scrutinise and debate this legislation.	10./*t10.txt*	Brexit

U korpusu 10docs je deset tekstova, pri čemu tekstovi 2, 3 i 4 govore o istoj temi, tekstovi 7, 8 i 10 o drugoj temi, tekst 1 ima tematskih dodirnih točaka s grupacijama 2, 3, 4 i 7, 8, 10, ali je ipak dosta različit, dok je tekst 9 tematski potpuno drukčiji od ostalih. Prilikom modeliranja metode DLPDM, programiranja i utvrđivanja podrazumijevajućih vrijednosti hiperparametara bio je iznimno koristan. Iako korpus 10docs nije korišten za službenu evaluaciju metoda, pogodan je i za prototipiranje konačnog izlaza sustava za otkrivanje parafraziranja, što je prikazano u sljedećoj tablici 5[1].

Tablica 5. Rezultati parafraziranosti između tekstova korpusa 10docs

Par tekstova	Sličnost
t02.txt - t04.txt	9.5% u t02.txt ili 14.3% u t04.txt
t02.txt - t03.txt	4.8% u t02.txt ili 18.8% u t03.txt
t03.txt - t04.txt	31.2% u t03.txt ili 11.9% u t04.txt
t07.txt - t10.txt	5.6% u t07.txt ili 10.0% u t10.txt
t08.txt - t10.txt	17.9% u t08.txt ili 25.0% u t10.txt

U tablici 5 vrijednosti sličnosti za preostale parove dokumenata nisu navedene jer sustav ne prepoznaje sličnost za sve ostale parove dokumenata.

4.3.2. Korpus CS

Clough i Stevenson (CS) je korpus parafraziranih tekstova razvijen za istraživanje i evaluaciju metoda za detekciju plagijata (Clough i Stevenson, 2011). Sastoji se od 100 kratkih tekstova raznih vrsta koje su autori kategorizirali prema razini parafraziranja na neparafrazirane (s oznakom „non”), direktno kopirane (s oznakom „cut”), lagano parafrazirane (s oznakom „light”) i jako parafrazirane (s oznakom „heavy”). Pored navedenih oznaka koristi se i ona za originalne tekstove („orig”). Pet originalnih tekstova preuzeto je iz akademskih javnih izvora. Odgovori su modificirani kako bi odgovarali različitim kategorijama plagiranja. Autori su koristili samo plagirane odgovore koji su prema *Turnitinu* imali najmanje 50% sličnosti s izvornim materijalom. Odbacili su odgovore koji su bili

1 Iako ova tablica djeluje kao strano tijelo, s obzirom na to da će rezultati tek biti prikazani u nastavku, o korpusu 10docs u nastavku više neće biti govora, pa ni prilike za uvid u sažeti prikaz gotovoga programskog sustava.

prekratki (manje od 50 riječi) ili predugi (više od 500 riječi).

Tablica 6. Primjer teksta i njegovih oblika parafraziranja iz korpusa CS

Tekst	Oznaka
In probability theory, Bayes' theorem (often called Bayes' law after Rev Thomas Bayes) relates the conditional and marginal probabilities of two random events. It is often used to compute posterior probabilities given observations. For example, a patient may be observed to have certain symptoms. Bayes' theorem can be used to compute the probability that a proposed diagnosis is correct, given that observation. (See example 2) As a formal theorem, Bayes' theorem is valid in all common interpretations of probability. However, it plays a central role in the debate around the foundations of statistics: frequentist and Bayesian interpretations disagree about the ways in which probabilities should be assigned in applications. Frequentists assign probabilities to random events according to their frequencies of occurrence or to subsets of populations as proportions of the whole, while Bayesians describe probabilities in terms of beliefs and degrees of uncertainty. The articles on Bayesian probability and frequentist probability discuss these debates in greater detail.	**orig**
Bayes' theorem relates the conditional and marginal probabilities of events A and B, where B has a non-vanishing probability: P(A|B) = \frac{P(B | A)\, P(A)}{P(B)}. Each term in Bayes' theorem has a conventional name: * P(A) is the prior probability or marginal probability of A. It is "prior" in the sense that it does not take into account any information about B. * P(A|B) is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B. * P(B|A) is the conditional probability of B given A. * P(B) is the prior or marginal probability of B, and acts as a normalizing constant. Intuitively, Bayes' theorem in this form describes the way in which one's beliefs about observing 'A' are updated by having observed 'B'.	
Baye’s theorm in connection with conditional probabilities is of fundamental importance, since it permits a calculation of PROB(AB) from PROB(BA). Statistical information that is often gathered in great volume can therefore be avoided	**non**
In probability theory, the prior and conditional probabilities of two random events are related by Bayes' theorem. The theorem is often used when we have observations and wish to compute posterior probabilities. For example, given an observation that a patient is seen to have certain symptoms, we can use Bayes' theorem to compute the probability that a suggested diagnosis is correct. P(A) is the prior probability of A. P(A|B) is the conditional probabilty of A given B. P(B|A) is the conditional probabilty of B given A. P(B) is the prior probability of B, and must be non-zero. Bayes' theorem is given by P(A|B) = (P(B|A)P(A))/(P(B)).	**heavy**
Bayes' theorem (often called Bayes' law) connects the conditional and marginal probabilities of two arbitrary events. One of its uses is calculating posterior probabilities given observations. Bayes' theorem plays a key role in the debate around the principles of statistics: frequentist and Bayesian interpretations disagree about the ways in which probabilities should be assigned in applications. Bayes' theorem is useful in evaluating the result of drug tests. If a test can identify a drug user 99% of the time, and can identify a non-user as testing negative 99% of the time, it may seem to be a relatively accurate test. However, Bayes' theorem will reveal the flaw	**light**

Tekst	Oznaka
that despite the apparently high accuracy of the test, the probability that an employee who tested positive actually did use drugs is only about 33%.	
In probability theory; Bayes theorem (often called Bayes law after Rev Thomas Bayes) relates the conditional and marginal probabilities of two random events. It is used to compute posterior probabilities given observations. For example; a person may be observed to have certain symptoms. Bayes theorem can be used to compute the probability that a proposed diagnosis is correct. As a formal theorem Bayes theorem is valid in all common interpretations of probability. However, it plays a central role in the debate around the foundations of statistics: frequentist and Bayesian interpretations disagree about the ways in which probabilities should be assigned to each other. Bayesians describe probabilities in terms of beliefs and degrees of uncertainty, While frequentists assign probabilities to random events according to their frequencies of occurrence or to subsets of populations as proportions of the whole. The articles on Bayesian probability and frequentist probability discuss these debates in detail.	**cut**

* Između svih kategorija parafraziranja iz korpusa (non, light, heavy, cut) i originalnog teksta (orig), u tablicu su uvršteni najkraći tekstovi.

4.3.3. Korpus MSRP

Microsoft Research Paraphrase Corpus (MRPC) je korpus koji se sastoji od 5801 parova rečenica prikupljenih iz novinskih članaka (Dolan i Brockett, 2005). Svaki par je označen je li ili nije parafraziran. Taj je korpus standardni resurs za istraživanje i razvoj modela za obradu prirodnog jezika. Korpus je podijeljen na skup za treniranje (4076 parova rečenica) i skup za testiranje (1725 parova rečenica). Oznake „1" ili „0" označavaju parafrazirano ili neparafrazirano, a dodijelili su ih ljudi, što načelno osigurava visoku kvalitetu podataka, no u istraživanju je identificirano puno primjera krivih oznaka (vidi *Dodatak: Primjeri krivih oznaka korpusa MSRP*).

Tablica 7. Primjeri parova rečenica iz korpusa MSRP s oznakama parafraziranosti

Tekst 1	Tekst 2	Oznaka
PCCW's chief operating officer, Mike Butcher, and Alex Arena, the chief financial officer, will report directly to Mr So.	Current Chief Operating Officer Mike Butcher and Group Chief Financial Officer Alex Arena will report to So.	1
The world's two largest automakers said their U.S. sales declined more than predicted last month as a late summer sales frenzy caused more of an industry backlash than expected.	Domestic sales at both GM and No. 2 Ford Motor Co. declined more than predicted as a late summer sales frenzy prompted a larger-than-expected industry backlash.	1
According to the federal Centers for Disease Control and Prevention (news - web sites), there were 19 reported cases of	The Centers for Disease Control and Prevention said there were 19 reported cases of measles in the United States in	1

Tekst 1	Tekst 2	Oznaka
measles in the United States in 2002.	2002.	
The settling companies would also assign their possible claims against the underwriters to the investor plaintiffs, he added.	Under the agreement, the settling companies will also assign their potential claims against the underwriters to the investors, he added.	1
A tropical storm rapidly developed in the Gulf of Mexico Sunday and was expected to hit somewhere along the Texas or Louisiana coasts by Monday night.	A tropical storm rapidly developed in the Gulf of Mexico on Sunday and could have hurricane-force winds when it hits land somewhere along the Louisiana coast Monday night.	0
The company didn't detail the costs of the replacement and repairs.	But company officials expect the costs of the replacement work to run into the millions of dollars.	0
Air Commodore Quaife said the Hornets remained on three-minute alert throughout the operation.	Air Commodore John Quaife said the security operation was unprecedented.	0
The broader Standard & Poor's 500 Index <.SPX> was 0.46 points lower, or 0.05 percent, at 997.02.	The technology-laced Nasdaq Composite Index .IXIC was up 7.42 points, or 0.45 percent, at 1,653.44.	0

***** Primjeri su odabrani slijedno iz korpusa, po četiri prvih rezultata s oznakom 1 i 0.

4.3.4. Korpus P4PIN

Paraphrase for Plagiarism Including Negatives examples (P4PIN) korpus se sastoji od 6708 rečenica koje su parafrazirane na način da su uključeni i pozitivni i negativni primjeri parafraziranja, tj. korpus sadrži rečenice koje su parafrazirane tako da zadržavaju izvorno značenje (pozitivni primjeri) i one koje mijenjaju značenje ili sadrže netočnosti (negativni primjeri) (Sánchez-Vega i sur., 2019). Na taj se način ostvaruje bolje treniranje modela za detekciju plagijata jer model može naučiti razliku između autentičnih i neautentičnih parafraziranja. Izvorni tekstovi dobiveni su iz akademskih radova, novinskih članaka i *online*-foruma. Oznake „1" ili „0", koje označavaju parafrazirano ili neparafrazirano, dodijelili su ljudi.

Tablica 8. Primjeri parova rečenica iz korpusa P4PIN s oznakama parafraziranosti

Izvorni tekst	Parafrazirani tekst	Oznaka
In order to move us, it needs no reference to any recognised original. It is there in virtue of the vesture of humanity in which it is clothed, and makes its appeal at once and directly. It is usual to speak of all the fine arts as imitative arts.	All art is imitation of nature. One does not need to recognize a tangible object to be moved by its artistic representation. Here by virtue of humanity's vestures, lies its appeal.	1
He has selected a personage for his drama with whom a certain fate is so indissolubly associated, that it is impossible to think of her without recalling it to mind; and this ineffaceable trait in her history he has attempted, for the time, to obliterate from our memory.	He has selected a personage for his drama with whom a certain fate is so indissolubly associated, that it is impossible to think of her without recalling it to mind; and this ineffaceable trait in her history he has attempted, for the time, to obliterate from our memory.	1
This Query is, of course, intimately connected with the much-disputed question of the origin of the Pointed Style itself. But yet I imagine that the application of the term "Gothic" may be found to be quite distinct, in its origin, from the first rise of the Pointed Arch.	This question is linked closely to the often-debated issue of the Pointed Style's beginnings. Still, in my opinion, the use of "Gothic" might well have origins unconnected to the emergence of the pointed arch.	1
Having thus laid down and discussed the principles which ought to regulate the constitution of the federal judiciary, we will proceed to test, by these principles, the particular powers of which, according to the plan of the convention, it is to be composed.	Since the principles regulating the constitution have already been established and talked about, next will be an examination of the specific powers it's supposed to have, as per the way the convention set it up.	1
The imitation of the drama is not that of any specific original it is a mimic scene, having human nature for its type. It has a life of its own, constructed from the materials which the records and observations of real life have supplied.	All art is imitation of nature. One does not need to recognize a tangible object to be moved by its artistic representation. Here by virtue of humanity's vestures, lies its appeal.	0
characters on the stage, they are not imitations of any definite originals, but they are invested with certain accidents and attributes of humanity, which give them at once the interest we feel in them, and set them living and moving in their own mimic world.	All art is imitation of nature. One does not need to recognize a tangible object to be moved by its artistic representation. Here by virtue of humanity's vestures, lies its appeal.	0
that he had followed faithfully the historical narrative, so neither do we impose upon him a very close adherence to it. We censure the course which Schiller has here pursued, not because he has marred history, but because he has marred his own poem. The objection lies entirely within the boundary of his own art.	He has selected a personage for his drama with whom a certain fate is so indissolubly associated, that it is impossible to think of her without recalling it to mind; and this ineffaceable trait in her history he has attempted, for the time, to obliterate from our memory.	0

Izvorni tekst	Parafrazirani tekst	Oznaka
By this procedure, the imagination of the reader is divided and distracted. The picture presented by the poet is and is not a portrait of the historical figure which lives in our recollection. There are many points of resemblance but the chief is omitted. And we always feel that it is omitted for history here is too strong for the poet he cannot expel her from the territory he wishes to enclose for himself.	He has selected a personage for his drama with whom a certain fate is so indissolubly associated, that it is impossible to think of her without recalling it to mind; and this ineffaceable trait in her history he has attempted, for the time, to obliterate from our memory.	0

* Primjeri su odabrani slijedno iz službenih oznaka, po četiri prvih rezultata s oznakom 1 i 0

4.3.5. Korpusi VMEN i VMENAIA

Novi, tijekom istraživanja načinjen korpus *Vrbanec Meštrović English* (VMEN) sastoji se od 100 parova kratkih tekstova i njihovih parafraziranih verzija. Kratki su tekstovi sažeci znanstvenih članaka na engleskom jeziku, preuzeti s *Hrčka*, baze hrvatskih znanstvenih časopisa (HRČAK, 2021). Za svaki originalni sažetak članka ručno je kreiran njegov parafraziran ekvivalent, što je rezultiralo konačnim VMEN korpusom koji sadrži 200 tekstova. Ručno parafraziranje provedeno je iz razloga što su prethodno iskušavani različiti automatizirani postupci, poput programski automatiziranoga višestrukog prevođenja između svjetskih jezika i korištenja nekoliko *online*-alata za parafraziranje, ali su rezultati parafraziranja bili nezadovoljavajući (ljudska ocjena „kvalitete" teksta). kod automatiziranoga višestrukog prevođenja korištene su tri platforme, uporabom njihova API sučelja: *Google Translator*, *Yandex* i *WordAI*. Transformacija koja je trebala rezultirati parafraziranjem uključivala je uzastopno prevođenje s engleskoga na njemački, s njemačkoga na španjolski te sa španjolskoga ponovo na engleski. Što se tiče *online*-alata za parafraziranje, iskušano ih je nekoliko, također s nezadovoljavajućim rezultatima. Nezadovoljavajući rezultati podrazumijevaju da su parafrazirajući tekstovi nerazumljivi, teško razumljivi ili nisu semantički sukladni. U konačnici je parafraziranje izvedeno ručno, parafraziranjem izvornih tekstova te potom njihovim lektoriranjem. Istodobno je za buduća istraživanja načinjen i korpus VMHR na hrvatskom jeziku koji čine sažeci iz korpusa VMEN prevedeni s engleskoga jezika. VMHR pored sažetaka sadrži i njihove parafraze na hrvatskom jeziku.

S obzirom na to da kod usporedbe rečenica VMEN nije imao potrebnu strukturu koja bi omogućavala 1:1 preslikavanje originalnih rečenica i njihovih parafraziranih ekvivalenata,

tj. *ground true* oznake, korpus je u pogledu parafraziranih tekstova ponovo kreiran, pri čemu su pomogli generativni jezični modeli koji su u međuvremenu postali javno dostupni (Google, 2023; OpenAI, 2024a). Rečenice su poravnate tako da svaka rečenica iz dokumenta s oznakom „original“ po svome slijedu odgovara rečenici iz dokumenta s oznakom „paraphrase“ (i obrnuto). Taj korpus nazvan je VMENAIA (engl. *Vrbanec i Meštrović English Paraphrase Corpora, AI assisted, Aligned*). Obje verzije korpusa na engleskom jeziku korištene su u različitim fazama ovoga istraživanja i dostupne su u otvorenom pristupu na https://vrbanec.com/corpora, dok će korpus na hrvatskome jeziku biti objavljen prilikom predstavljanja rezultata budućih istraživanja. Može se reći da je VMENAIA **kvalitetan korpus parafraziranih tekstova** jer je višestruko parafraziran (ručno te uz pomoć dvaju jezičnih modela), prikrivenog tipa parafraziranja metodom parafraziranja, s velikom koncentracijom istih tema s obzirom na pripadnost istoj domeni računalnih i informacijskih znanosti, tako da je pronalaženje parafraziranih dokumenata i rečenica između dokumenata iz iste domene težak zadatak, potpuno usporediv s pronalaženjem prikrivenoga plagiranja metodom parafraziranja u stvarnom svijetu. K tome je korpus nastao iz akademskih radova, preciznije njihovih sažetaka, poprilično nejednakih veličina, pa je stoga dobar test za bilo koji klasifikator kojem je cilj pronalaženje akademskih plagijata nastalih prikrivenim plagiranjem.

Tablica 9. Primjer para tekstova korpusa VMENAIA

Izvorni tekst	Parafrazirani tekst
The success of a tourist destination on the market in the time of advanced technology and high traveller expectations is becoming hard to achieve without a solid Internet presence. Communication and marketing strategies in a Digital era have to satisfy the modern man who wants to be approached in a unique way, and catered to in accordance to his specific needs. This is especially pronounced in young people as the most demanding consumer group, since they base their choice of tourist destination on the information gathered through various communication channels, most of all through social networks. This paper analyses the connection between social networks as contemporary tool of transferring information and the process of decision making in choosing a tourist destination in student population. The research in this paper aims to examine ways of using social networks by student population when choosing a vacation destination. It more specifically looks to decipher differences in the role of social networks in choosing a tourist destination taking age and financial status of examinees into account. Finally it analyses examinees' habits of sharing their vacation experience with other users of social networks. The research has been conducted on a convenience sample of 100 examinees by method of a survey among the student of Specialist professional graduate studies Marketing and communications Zagreb School of Business.	In the contemporary landscape of sophisticated technology and discerning travelers, the triumph of a tourist spot in the marketplace is increasingly reliant on a robust online presence. In today's landscape, effective communication and marketing strategies need to recognize and address the individual's preference for feeling valued and receiving offers that cater to their unique needs. This trend is particularly evident among young people, the most exacting consumer demographic, who predominantly base their travel choices on information gleaned from diverse communication channels, notably social media platforms. The present study investigates the correlation between social media as a contemporary conduit for information dissemination and the decision-making process involved in selecting a travel destination among the student population. The research endeavors to scrutinize the ways in which students utilize social media when choosing their vacation spots. The study focuses on uncovering how the influence of social media on travel destination selection varies based on factors like age and income levels. Furthermore, the study analyzes the respondents' proclivity for sharing their travel encounters with fellow social media users. The investigation was carried out on a convenience sample of 100 respondents through a survey administered to students enrolled in the Specialist Professional Graduate Studies in Marketing and Communications at the Zagreb School of Business.

* Prvi par tekstova iz korpusa

4.3.6. Korpus Webis

Webis-11 je korpus parafraziranih tekstova koji sadrži 7859 parova tekstova (u zasebnim tekstnim dokumentima) koji su semantički slični, ali su različito formulirani (Burrows i sur., 2013). Neke su parafraze načinili ljudi, neke su stvorene automatski pomoću algoritama za parafraziranje, dok su preostale stvorene kombinacijom ručnoga i automatskog pristupa. Korpus je predviđen za razvoj i evaluaciju algoritama za parafraziranje i prepoznavanje parafraziranja. Zanimljivo je da je 335 tekstova odnosno dokumenata potpuno

prazno, tj. bez sadržaja. Oznake „1” ili „0” označavaju parafrazirane ili neparafrazirane parove tekstova.

Tablica 10. Primjeri parova rečenica iz korpusa Webis

Izvorni tekst	Parafrazirani tekst	Oznaka
M. Comte would not advise so irrational a proceeding. But M. Comte has himself a constructive doctrine; M. Comte will give us in exchange—what? The Scientific Method! We have just seen something of this scientific method.	Even M. Comte would spurn such irrational reasoning. However, M. Comte adheres himself to a fruitful belief, one which he will offer us instead - the Scientific Method! This scientific method has, in fact, just been observed.	1
Without enumerating all the modern authors who hold this view, we will quote a work which has just appeared with the imprimatur of Father Lepidi, the Master of the Sacred Palace, in which we find the two following theses proved: 1.	Just without specifying the current writers who have this view, we will proceed with the work just came with the impremature of Father Lepidi, the Master of Sacred palace, which proves the following theses proved: 1.	1
Therefore, a person should search his actions and repent his transgressions previous to the day of judgment. In the month of Elul (September) he should arouse himself to a consciousness of the dread justice awaiting all mankind.	As such, a person should analyze what he did and be sorry for his mistakes before judgment day. In September, also referred to as Elul, he should force himself of the frightening justice that awaits all humans.	1
"I have heard many accounts of him," said Emily, "all differing from each other: I think, however, that the generality of people rather incline to Mrs. Dalton's opinion than to yours, Lady Margaret." "I can easily believe it.	Emily said, "I have heard many different things about him; however, most people trust Mrs. Dalton's beliefs more then they do yours, Lady Margaret, myself included."	1
"I have heard many accounts of him," said Emily, "all differing from each other: I think, however, that the generality of people rather incline to Mrs. Dalton's opinion than to yours, Lady Margaret." "I can easily believe it.	"I have heard many accounts of him," said Emily, "all different from each other: I think, however, that the generality of the people rather inclined to the view of Ms Dalton to yours, Lady Margaret." That I can not believe.	0
"Gentle swain, under the king of outlaws," said he, "the unfortunate Gerismond, who having lost his kingdom, crowneth his thoughts with content, accounting it better to govern among poor men in peace, than great men in danger."	"gentle swain,under the king of outlaws", said he ,"the fortunate gerismond,who having lost his Kingdom,crowneth his thoughts with the content,accounting it better to govern among poor men in peace, the great men in danger"	0
[Greek: DEMOSTHENOUS O PERI TÊS PARAPRESBEIAS LOGOS.] DEMOSTHENES DE FALSA LEGATIONE. By RICHARD SHILLETO, M.A., Trinity College, Cambridge. Second Edition, carefully revised. Cambridge: JOHN DEIGHTON. London: GEORGE BELL.	[Hellene: DEMOSTHENOUS O PERI TÊS PARAPRESBEIAS LOGOS.] Speechmaker DE FALSA LEGATIONE. By RICHARD SHILLETO, M.A., Divine College, Metropolis. Indorse Edition, carefully revised. Metropolis: Evangel DEIGHTON. Author: Martyr Push.	0

Izvorni tekst	Parafrazirani tekst	Oznaka
[Greek: DEMOSTHENOUS O PERI TÊS PARAPRESBEIAS LOGOS.] DEMOSTHENES DE FALSA LEGATIONE. By RICHARD SHILLETO, M.A., Trinity College, Cambridge. Second Edition, carefully revised. Cambridge: JOHN DEIGHTON. London: GEORGE BELL.	[Greek: DEMOSTHENOUS O PERI TES PARAPRESBEIAS LOGOS.] DEMOSTHENES DE FALSA LEGATIONE. By RICHARD SHILLETO, M.A., Trinity College, Cambridge. Second Edition, carefully revised. Cambridge: JOHN DEIGHTON. London: GEORGE BELL.	0

* Primjeri su odabrani slijedno iz korpusa, po četiri prva rezultata s oznakom 1 i 0

4.3.7. Usporedba obilježja korpusa

Korpusi se razlikuju u svakom pogledu: broju tekstova, broju riječi (u cijelom korpusu i pojedinačnim tekstovima), maksimalnom i prosječnom broju riječi u tekstovima, varijanci, ekstremnim vrijednostima (engl. *outliers*). Dijagrami na slici 12. prikazuju distribuciju riječi korištenih korpusa, numerički prikazanih u tablici 11. *Box-plot* dijagrami ilustriraju distribucije broja riječi u korpusima, a kružići označavaju ekstremne vrijednosti. *VMEN (reduciran)* i *Webis (reduciran)* imaju točno 1000 riječi kao maksimalan broj riječi. To je stoga što njihova aritmetička sredina uvećana za dvije standardne devijacije nije prelazila 1000 riječi. Točan broj riječi za smanjenje odstupanja izračunat je pomoću jednadžbe

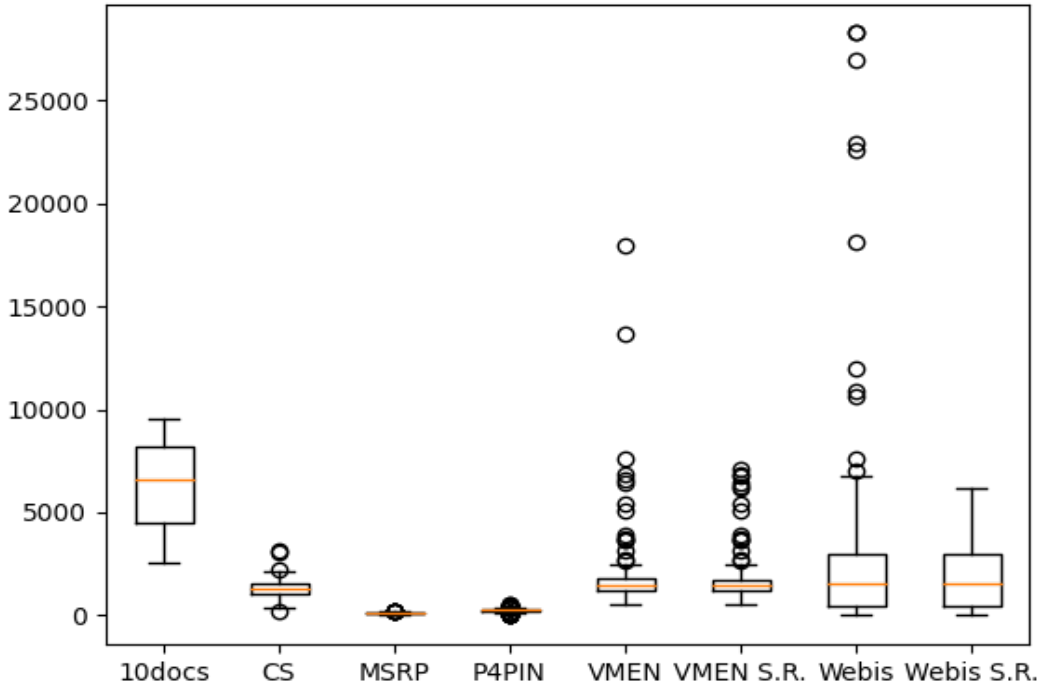
$$Max_{num} = Max\left(Activator\,;\, Mean + 2 \cdot SigmaValue\right) \tag{45}$$

gdje je *Activator* broj riječi potreban za aktivaciju skraćivanja teksta – parametar podešen na 1000, *Mean* je srednja vrijednost broja riječi tekstova pripadnoga korpusa, a *SigmaValue* (σ) je standardna devijacija broja riječi tekstova pripadnoga korpusa – mjera raspršenosti podataka oko njihove srednje vrijednosti (aritmetičke sredine). U eksperimentima s navedenim dvama korpusima korišteni su reducirani korpusi umjesto originalnih, tako da dimenzije jezičnih modela dubokog učenja za originalne korpuse nisu izračunate jer nisu ni korištene („-" vrijednosti).

Tablica 11. Značajke korpusa

Korpus	Oznake parafraziranja	Klase (%)	Broj tekstova	Broj riječi	Maksimalan broj riječi	Aritmetička sredina	Standardna devijacija	Dimenzija
10docs	nema oznaka	-	10	10465	1611	1046.5	376.2	38
CS	orig, non, light, heavy, cut	5, 38, 19, 19, 19	100	21440	529	214.4	77.97	73
MSRP	0, 1	32, 68	10948	211206	34	19.29	5.17	188
P4PIN	0, 1	75, 25	6708	299050	90	44.58	9.69	182
VMEN	0, 1	99, 1	200	55201	2632	276	255.89	-
VMEN (R)	0, 1	99, 1	200	52296	1000	261.48	157.34	93
Webis	0, 1	48, 52	*15718	4928055	4993	320.34	272.42	-
Webis (R)	0, 1	48, 52	15383	4893147	1000	318.09	259.82	223

*Uključujući 335 praznih datoteka, (R) = reduciran

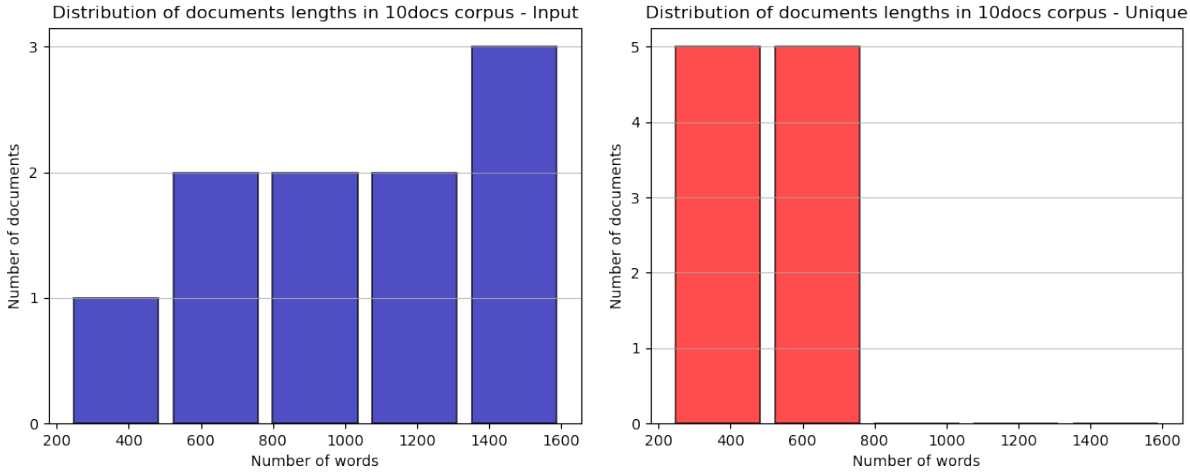


Slika 12. Distribucija broja riječi u korpusima (*box-plot* dijagrami)

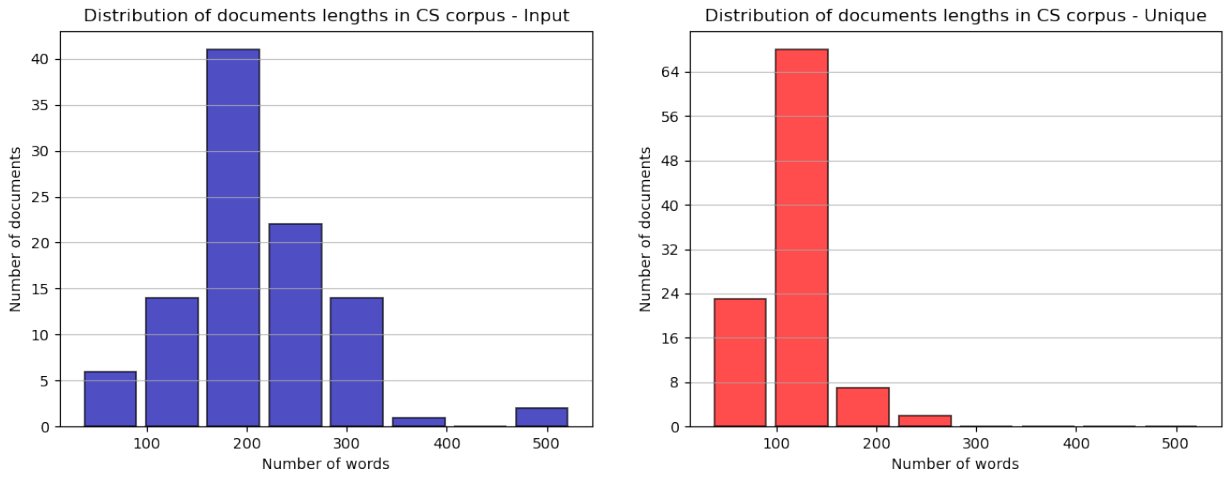
Slika 12 prikazuje distribuciju riječi u tekstovima korpusa. Svaki mali kružić na slici 12 predstavlja jednu ekstremnu vrijednost broja riječi teksta u korpusu kojem pripada, a njezin položaj na vertikalnoj osi predstavlja broj riječi te ekstremne vrijednosti. Mali korpus 10docs sadrži samo deset dokumenata srednje veličine, bez ekstremnih vrijednosti. Korpusi MSRP i P4PIN imaju nekoliko ekstremnih vrijednosti, no one nisu ni problematične (za računalnu obradu) s obzirom na to da su to dva rečenična korpusa, tj. njihovi su tekstovi u velikom broju slučajeva sastavljeni od samo jedne rečenice, a kada to i nije tako, riječ je o maksimalno tri rečenice koje bismo mogli aproksimirati kao jednu složenu. CS i VMEN imaju nekoliko ekstremnih vrijednosti, dok Webis ima značajan broj ekstremnih vrijednosti. Posljedica postojanja tekstova s ekstremnim brojem riječi u eksperimentima bila je velika potrošnja radne memorije tijekom izvođenja programa Python, što je rezultiralo intenzivnom upotrebom virtualne memorije. Ukupno, posljedica toga je iznimno dugo trajanje eksperimenata (reda veličine tjedana i mjeseci), a često i nemogućnost njegova završetka. Stoga se veličina onih tekstova s ekstremnim brojem riječi morala smanjiti na prikladnu veličinu (odsijecanjem njihovih završetaka). Dva hiperparametra korištena su za rješavanje

problema ekstremnih vrijednosti, a njihove vrijednosti korištene su kao prag za odsijecanje završetaka teksta ovisno o srednjoj vrijednosti i standardnoj devijaciji korpusa. Na temelju rezultata eksperimenta i zdrave logike veličina ekstremno velikih tekstova ograničena je na dvije standardne devijacije više od njihove aritmetičke sredine, uz preduvjet (aktivacija ograničenja) da tekst mora sadržavati više od 1000 riječi (znajući da je 250–300 engleskih riječi ekvivalentno jednoj kartici teksta). Dakle, ako tekst ima više od 1000 riječi te je broj tih riječi veći od srednje vrijednosti uvećane za dvije standardne devijacije, ostatak riječi nekog teksta iznad tako izračunate vrijednosti se reže, a korpus se pretvara u reducirani korpus. Numeričke vrijednosti tablice 11 upotpunjuju elementi prikazani na *box-plot* dijagramima na slici 12 koji daju vizualni uvid u distribuciju broja riječi koje čine korpuse, a posebno daju uvid u *outlier* vrijednosti.

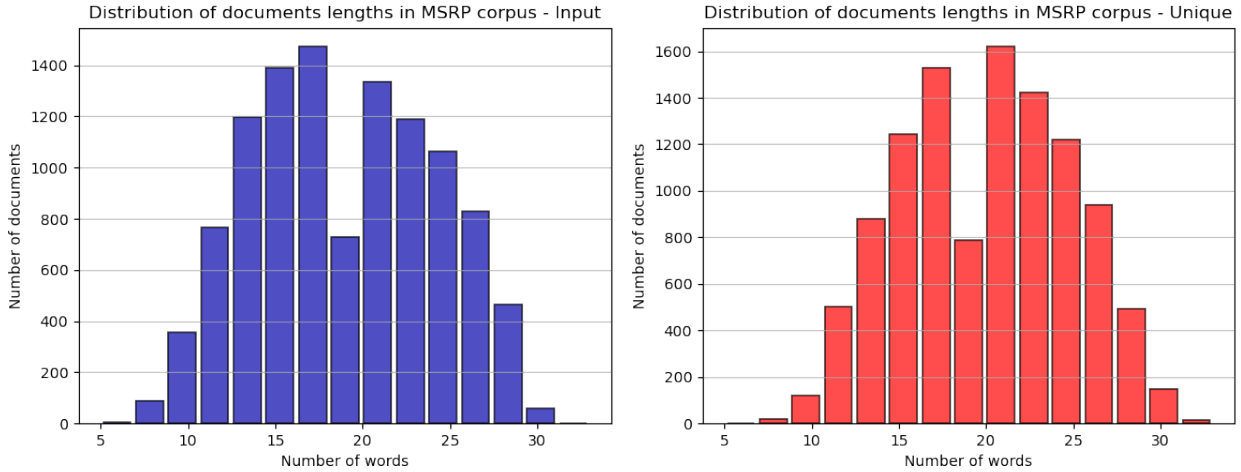
Distribucije broja riječi dokumenata u korpusima prikazuju i histogrami na slikama 13-18. Pri tome svaki korpus ima svoj ulazni, izvorni oblik („Input“ histogram), histogram brojeva jedinstvenih riječi („Unique“ histogram), a dva korpusa, VMEN i Webis, dodatno imaju i histograme njihovih reduciranih oblika („Reduced and Cleaned“ histogram).



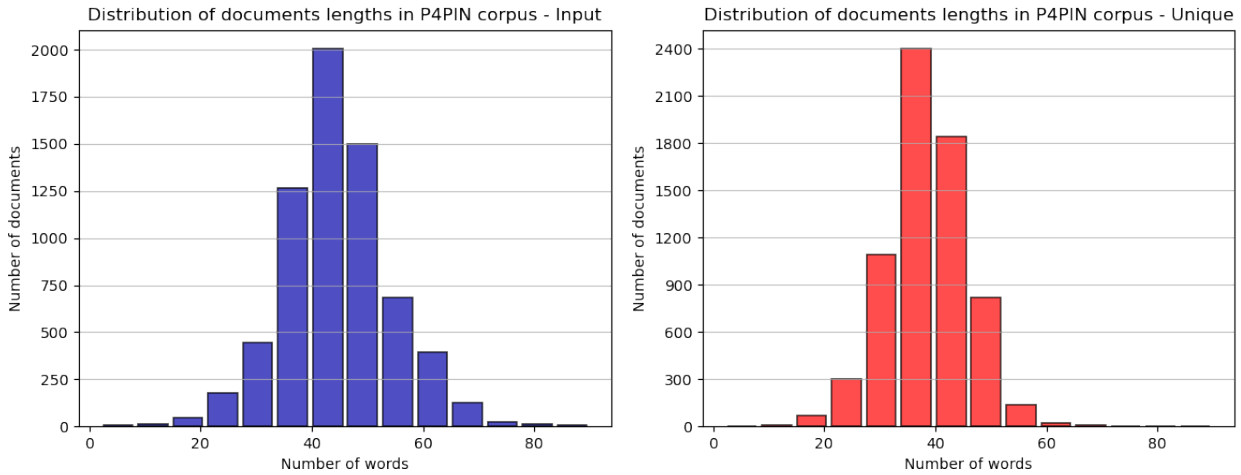
Slika 13. Distribucija broja riječi korpusa 10docs



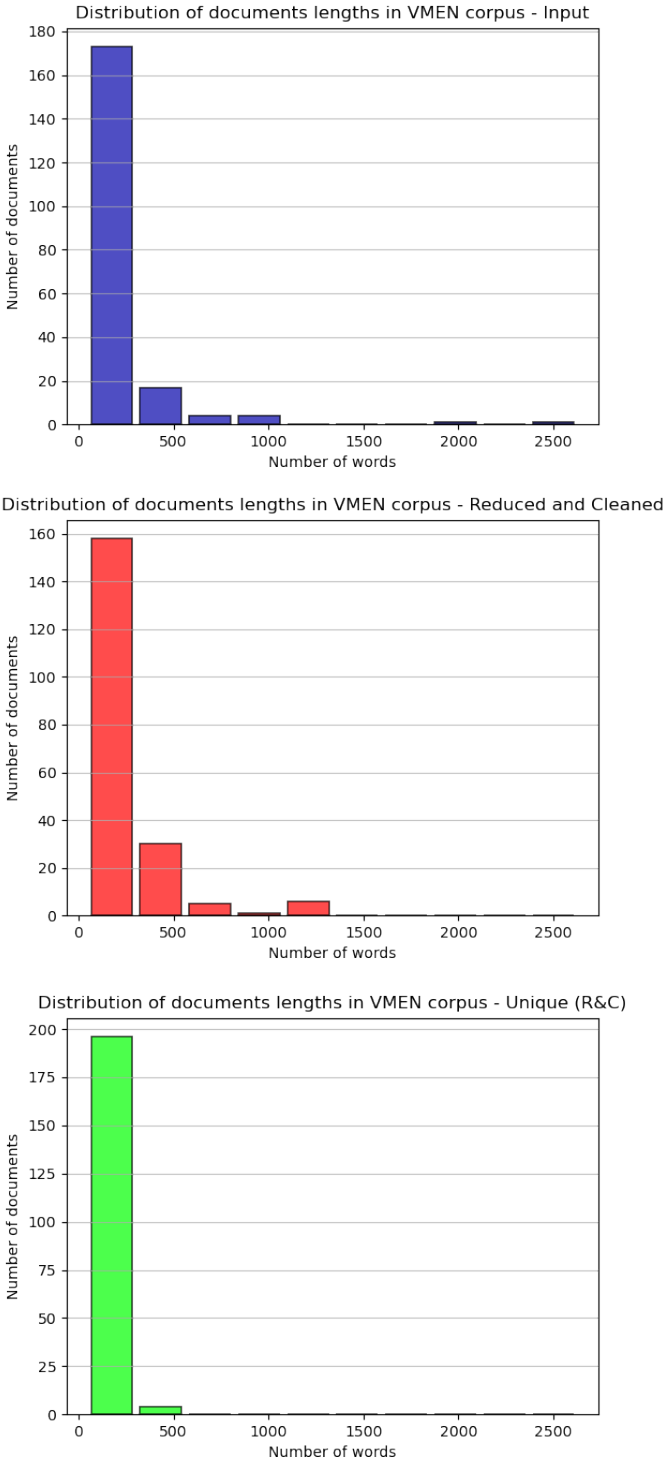
Slika 14. Distribucija broja riječi korpusa CS



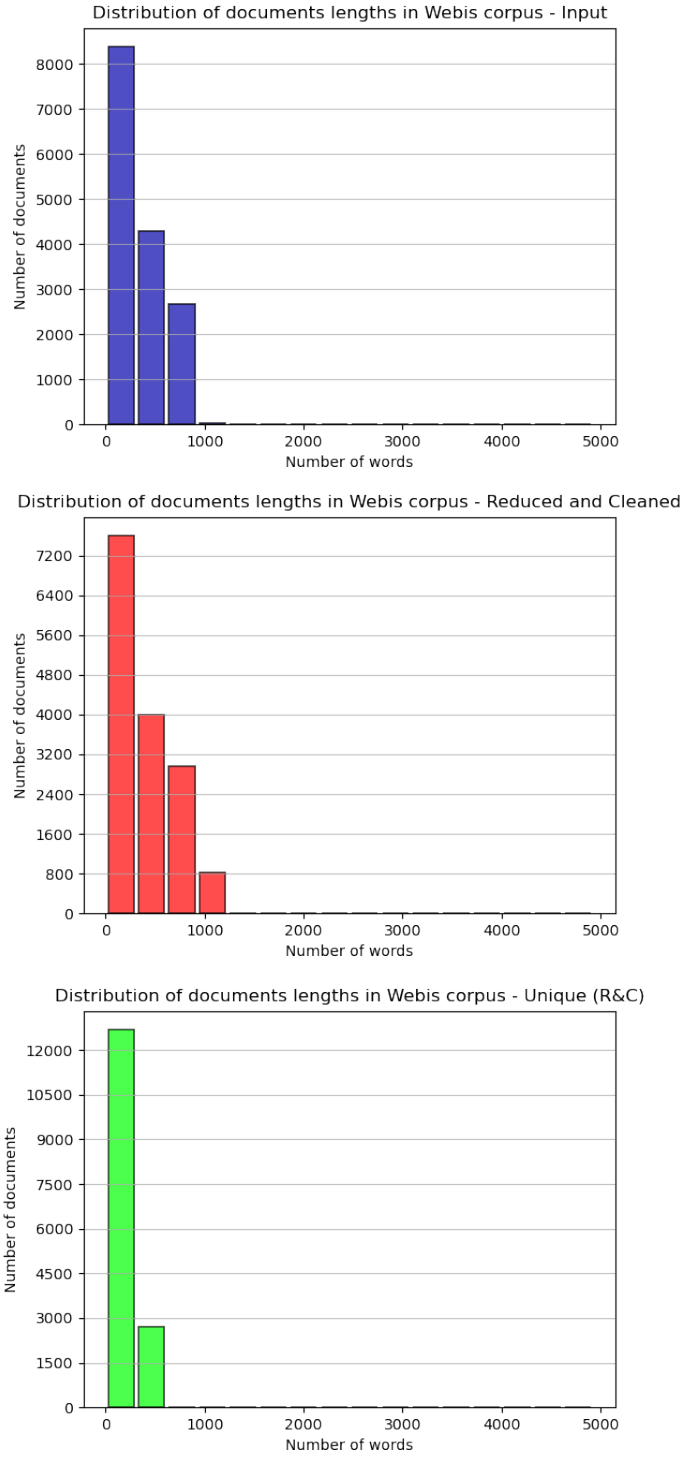
Slika 15. Distribucija broja riječi korpusa MSRP



Slika 16. Distribucija broja riječi korpusa P4PIN



Slika 17. Distribucija broja riječi korpusa VMEN



Slika 18. Distribucija broja riječi korpusa Webis

5. Eksperimenti

U ovom poglavlju opisana je implementacija eksperimentalnih postupaka kroz koje su istraženi i utvrđeni pojedini elementi metode DLPDM. U prvom koraku implementirani su i analizirani različiti postupci iz područja NLP-a za pretprocesiranje tekstova dokumenata. Nakon toga proveden je sveobuhvatni niz eksperimenata koristeći ukupno 206 različitih modela i metoda za reprezentaciju teksta na šest korpusa označenih parafraziranih tekstova, uz dvije mjere sličnosti i tri mjere udaljenosti kako bi se precizno rangirali modeli prema F1-mjeri, i utvrdili najbolji jezični modeli i mjere sličnosti/udaljenosti za otkrivanje parafraziranja. U okviru provedenih eksperimenata, za svaku kombinaciju pristupa utvrđen je optimalni prag za binarizaciju rezultata u izračunu mjere sličnosti na razini dokumenata i mjere parafraziranja na razini rečenica. Kroz opisani niz eksperimenata koji su implementirani u programskome jeziku *Python* detaljno su istraženi elementi metode DLPDM, a to su: (i) postupci obrade teksta, (ii) modeli i metode za reprezentaciju teksta, (iii) mjere sličnosti te (iv) pragovi sličnosti. Na temelju rezultata provedenih eksperimenata, u konačnici su utvrđeni odabrani konkretni niz metoda obrade teksta ($\boldsymbol{O_{txt}}$), konkretni model ($\boldsymbol{M^*}$), mjera sličnosti (***sim***) te vrijednosti pragova sličnosti ($\boldsymbol{\theta^*}$ i $\boldsymbol{\theta^{**}}$) koji daju najbolje rezultate u zadatku detekcije parafraziranja.

U potpoglavlju *5.1. Priprema podataka* opisana je priprema podataka, odnosno implementacija i analiza postupaka za pripremu tekstova, a također i priprema dokumenata iz korpusa za treniranje modela. U potpoglavljima *5.2. Eksperimenti za utvrđivanje sličnosti na razini dokumenta* i *5.3. Eksperimenti za utvrđivanje parafraziranja na razini rečenica* detaljno su opisani eksperimenti provedeni nad cjelovitim tekstovima i nad rečenicama. Nakon toga je u potpoglavlju *5.4. Analiza složenosti* istražena i opisana složenost postupaka. Na kraju su u poglavlju *5.5. Tehničke specifikacije* opisane tehničke specifikacije u okviru kojih je provedena implementacija eksperimenata.

5.1. Priprema podataka

5.1.1. Eksperimenti s tehnikama za obradu i pripremu teksta

Prvi korak u radu s cjelovitim tekstovima dokumenata bio je pretprocesiranje, odnosno

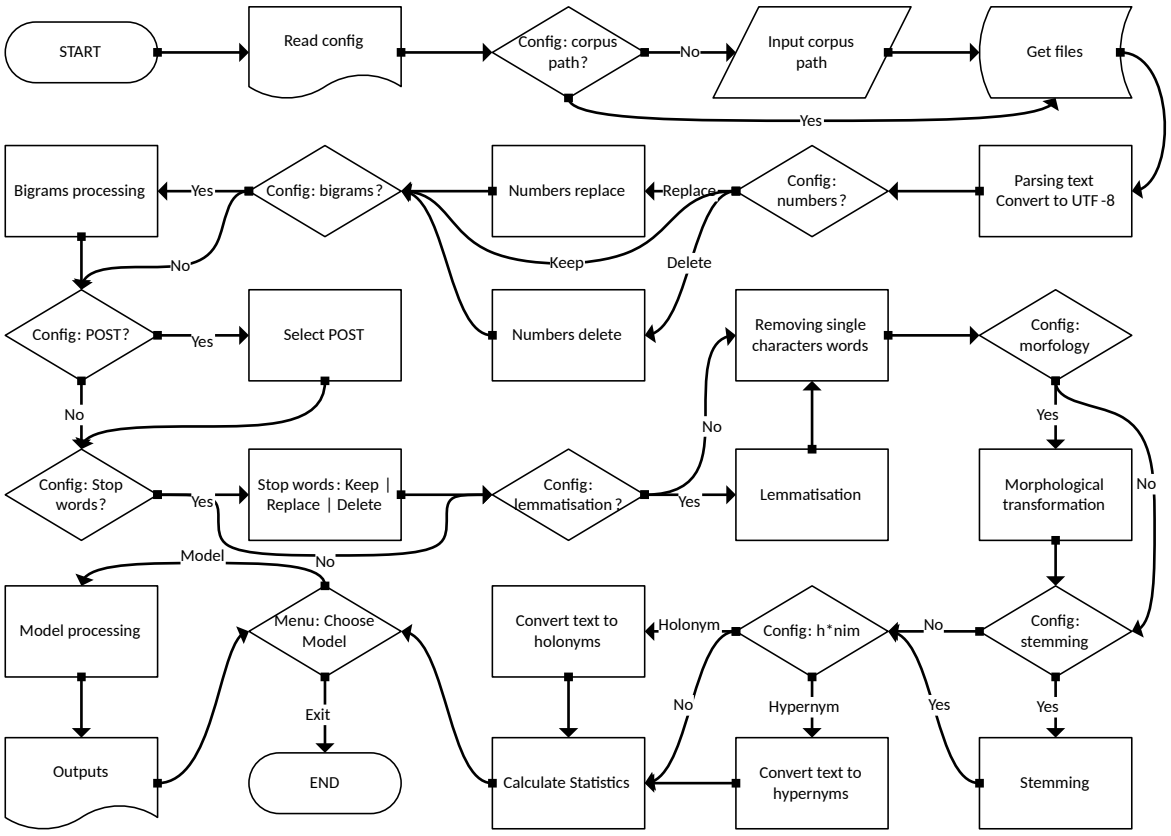
priprema teksta. Tekstovi su izvučeni iz različitih formata datoteka i predobrađeni korištenjem različitih metoda obrade prirodnog jezika prema metodologiji predstavljenoj u (Vrbanec i Meštrović, 2020). Parsiranjem dokumenata iz njih se ekstrahira neoblikovan tekst uzimajući u obzir različite potencijalne formate, jezike i kodiranja.

Nadalje, proveden je niz eksperimenata u kojima su isprobani različiti pristupi pretprocesiranja i pripreme teksta. Primjerice, isprobano je je li bolje izbaciti zaustavne riječi (*stop-words*) i brojeve iz teksta ili ne. Na temelju provedenih eksperimenata pokazalo se kako su rezultati bolji ako se ne koristi mnoštvo mogućnosti pretprocesiranja i pripreme teksta: korištenje n-grama riječi, uklanjanje ili zamjena brojeva, korištenje samo nekih tipova riječi (POST), uklanjanje ili zamjena zaustavnih riječi, lematizacija, uklanjanje riječi od samo jednog znaka, korištenje morfoloških transformacija *WordNeta*, korjenovanje, korištenje hiperonima i homonima kako je to prikazano na slici 19, dijagramom toka svih iskušanih načina obrade teksta. Zbog eksperimentalno dokazanih boljih rezultata tekstovi su dalje obrađeni tako da su primjerice brojevi i zaustavne riječi zadržani, za razliku od uobičajenog NLP postupka. Kao jedine učinkovite transformacije teksta pokazale su se njegova pretvorba u mala slova, tokenizacija i lematizacija. Za dobivanje riječi iz teksta korištena je *nltk Python* biblioteka u prvoj fazi istraživanja kod usporedbe cjelovitih tekstova. U drugoj fazi, kod usporedbe rečenica, umjesto *nltk* biblioteke korišteni su regularni izrazi kako za ekstrakciju rečenica tako i za riječi iz njih, jer su se oni pokazali pouzdanijim od *nltk* i *spacy* biblioteka. Riječi s jednim znakom ostavljene su u tekstovima jer je i opet takav pristup dao bolje rezultate. U USE, ELMo, BERT i Laser modelima korišten je nemodificirani obični tekst, ostavljajući čak i interpunkcijske znakove jer je to bio najbolji pristup za te modele.

Obrada teksta predstavlja važan dio istraživanja, te su isprobani različiti pristupi obrade, uključujući osnovne tehnike kao što su lematizacija, uklanjanje zaustavnih riječi, te normalizacija teksta (pretvaranje teksta u mala slova, uklanjanje interpunkcijskih i posebnih znakova, višestrukih razmaka, standardizacija brojeva).

Dijagram toka prikazan slikom 19 prikazuje različite postupke obrade teksta koji su isprobani prilikom eksperimenata (Vrbanec i Meštrović, 2020). Postupak je sveobuhvatan i fleksibilan te ovisi o konfiguraciji učitanih hiperparametara. Započinje učitavanjem inicijalne konfiguracije i datoteka korpusa, nastavlja se putem različitih faza transformacije i prilagodbe teksta, pa sve do konačne obrade i izračuna statističkih obilježja korpusa. Proces je potpuno prilagodljiv, s brojnim odlukama koje omogućuju korisniku definiranje specifičnih operacija,

čak i tijekom rada programa. Svaka odluka definirana je postavkama pripadajućih hiperparametra radi brzine preddefiniranog izvođenja brojnih metoda i nad nekoliko korpusa, što vodi do različitih grana obrade i omogućuje modularan pristup prilagođen različitim potrebama analize teksta. Završna faza obuhvaća izračun statistika i obradu rezultata za daljnju analizu ili upotrebu. Cjelokupan proces omogućuje visok stupanj prilagodljivosti i osigurava da se tekstni podaci pripreme na način koji omogućuje usporedbu te podešavanje hiperparametara u stanja koja omogućuju najbolje rezultate mjerene F1-mjerom.



Slika 19. Dijagram toka svih iskušanih obrada teksta

5.1.2. Podjela dokumenata u korpusima

Za potrebe provođenja eksperimenata bilo je potrebno unificirati oblike dokumenata u različitim korpusima. S obzirom na to da je programski kod za izradu prototipa koristio datoteke kao prirodni oblik usporedbe dvaju dokumenata, korpusi koji nisu imali datoteke već

zapise u jednoj datoteci, transformirani su u oblik datoteka. Nadalje, s obzirom na to da je prva evaluacija rezultata rađena na MSRP korpusu koji ima oznake – rezultate u *Excel* tablici, drugi su se rezultati (oznake) trebali također staviti u isti oblik *Excel* tablice. Istodobno je jako mnogo problema u prvoj fazi istraživanja bilo s kodiranjima teksta, i to iz razloga što je tada korišteni Python 2.7 koristio samo ASCII kod i nije podržavao univerzalne kodove, poput utf-8 ili utf-16. Poslije, prelaskom na Python 3, taj je problem gotovo iščeznuo, no to nije bio jedini problem kod obrade ulaznoga teksta. Na ulazu programa Python potrebno je bilo da on prihvaća bilo koji češći oblik tekstnog dokumenta *(txt, rtf, doc, docx, pdf, odt, html, json* i druge), pa su mnoge varijante oblikovanja teksta radile poteškoće, poput numeracije stranica, prijelaza stranica, prekida tijeka teksta zbog slika i tablica, tekst oblikovan u više stupaca, pdf-datoteke sa slikovnim sadržajem umjesto teksta (za koje je bila potrebna OCR obrada) i sl.

Nadalje, bilo je potrebno odrediti koji će se dio dokumenata iz korpusa koristiti za treniranje, a koji za testiranje modela. Korišten je standardni postupak podjele podatkovnih skupova, tj. korpusa: 80% skupa podataka nasumično je odabrano za treniranje, a 20% za testiranje. Za manje korpuse (tj. CS sa 100 tekstova i VMEN s 200 tekstova) morala se provesti unakrsna validacija (eksperimenata i njihovih rezultata) u pet ciklusa eksperimenata treniranja i testiranja, gdje je u svakom ciklusu 20% podatkovnog skupa (korpusa) nasumično odabrano za testni skup podataka (iz preostalog skupa onih koji nisu prethodno bili izabrani), a ostatak za skup podataka za treniranje. Vrijednosti pragova, graničnih vrijednosti za binarizaciju, dobivene su iz najboljih rezultata na eksperimentima treniranja te su primijenjene kao unaprijed definirane za eksperimente nad testnim dijelom korpusa. kod unakrsno validiranih korpusa rezultati pet ciklusa poslužili su za izračun aritmetičke sredine. Budući da je testni korpus 10docs korišten (samo) za izradu programa *Python* i podešavanje hiperparametara, kod drugih korpusa nije bilo potrebe izdvajati podskup za validaciju.

5.2. Eksperimenti za utvrđivanje sličnosti na razini dokumenta

U ovome poglavlju opisani su eksperimenti za određivanje sličnosti na razini dokumenta. U prvom dijelu opisan je tijek eksperimenata za detekciju najboljega modela za reprezentaciju teksta koji je proveden u dva ciklusa. U okviru prvoga potpoglavlja dodatno su opisani postupci analize mjere sličnosti te određivanje optimalne vrijednosti za prag sličnosti. Nakon toga kroz idućih nekoliko potpoglavlja predstavljeno je određivanje hiperparametara i

određivanje broja dimenzija vektorskoga prostora te je na kraju opisana tehnička specifikacija.

5.2.1. Modeli za reprezentaciju teksta

U prvoj fazi zadatka utvrđivanja parafraziranosti tekstova određivana je sličnost dokumenata, odnosno cjelovitih tekstova tijekom **dvaju ciklusa**: prvog sa 60 različitih metoda i drugoga sa 146 jezičnih modela temeljenih na arhitekturi transformera. Proveden je niz eksperimenata u okviru kojih je testirana široka lepeza metoda, od statističkih do kombiniranih jezičnih modela s mjerama sličnosti/udaljenosti.

U **prvom ciklusu** uspoređivane su dvije skupine pristupa za određivanje sličnosti tekstova: statistički temeljeni pristupi i pristupi temeljeni na dubokom učenju. Prva skupina bila je svojevrsna kontrolna skupina (engl. *baseline*) pristupa dajući referentne rezultate: *Term Frequency – Inverse Document Frequency* (Tf-Idf) (Salton i Buckley, 1988), *Latent Semantic Indexing* ili *Latent Semantic Analysis* (LSI) (Deerwester i sur., 1990), *Latent Dirichlet Allocation* (LDA) (Blei i sur., 2003), *Hierarchical Dirichlet Process* (HDP) (Teh i sur., 2006), *Random Projections* ili *Random Indexing* (RP) (Sahlgren, 2005), *LogEntropy* (LE) (M. D. Lee i sur., 2005), *Jaccard* (Jaccard, 1912), *Levenshtein* (Levenshtein, 1966), modificirana Greedy String Tilling (GST) ili *Scored Greedy String Tilling* (SGST) (Všianský, 2019) koja je korištena pod nazivom *Greedy Word Tilling* (GWT), što je izraz koji bolje opisuje tu mjeru sličnosti jer se ona u ovome slučaju primjenjuje na razini riječi, a ne stringa. Druga grupa pristupa su jezični modeli temeljeni na dubokom učenju i varijantama dubokih neuronskih mreža: *Word2Vec* (dvije varijante) (Mikolov i sur., 2013b), *Doc2Vec* (četiri varijante) (Le i Mikolov, 2014), *FastText* (Joulin i sur., 2016), *GloVe* (dvije varijante) (Pennington i sur., 2014), *USE* (Cer, Yang, Kong, Hua, Limtiaco, John, i sur., 2018), *ELMo* (Peters i sur., 2018), *Laser Embeddings* (Artetxe i Schwenk, 2019), *BERT* (Devlin i sur., 2018) i *Roberta* (Reimers i Gurevych, 2019).

Treba napomenuti da su se za *Word2Vec*, *Doc2Vec*, *GloVe* i *FastText* vektorske reprezentacije dobivale treniranjem neuronskih mreža na ulaznim korpusima. Za *USE*, *ELMo*, *BERT*, *Laser Embeddings* te ostale velike jezične modele uglavnom temeljene na arhitekturi transformera, vektorske reprezentacije preuzete su kao prethodno istrenirani modeli s interneta. Za takav zadatak modele nije bilo potrebno dodatno podešavati jer bi to bilo kontraproduktivno s obzirom na ideju da se otkrivanje parafraziranja može izvesti nad bilo

kojim skupom tekstova ili u bilo kojoj domeni, tako da bi forsiranje finog podešavanja za jednu domenu ili na nekom određenom podatkovnom skupu rušilo ideju pronalaženja parafraziranja općenito, a ne u nekoj (pod)domeni.

5.2.2. Eksperimenti s mjerama sličnosti

Nakon utvrđivanja najboljeg modela za vektorsku reprezentaciju teksta provedeni su eksperimenti u okviru kojih su uspoređene različite mjere sličnosti. Izračuni sličnosti između tekstova izvršeni su korištenjem vlastitog koda u *Pythonu* i uz pomoć nekoliko javno dostupnih programskih biblioteka. Istraživanje detaljnije prikazano u Vrbanec i Meštrović (2021a) nastojalo je utvrditi koji su parovi jezičnih modela i mjera sličnosti/udaljenosti najpogodniji za daljnja istraživanja otkrivanja plagijata nastalih parafraziranjem (Vrbanec i Meštrović, 2021a). Eksperimenti su korišteni za dobivanje sličnosti dokumenata prvobitno iz tri javno dostupna korpusa tekstnih dokumenata (CS, MSRP i Webis) te testni korpus 10docs, (korpusi su predstavljeni u poglavlju *4.3. Korpusi parafraziranih tekstova*). Kasnije je istraživanje (Vrbanec i Meštrović, 2023) uključilo dodatne korpuse (P4PIN, VMEN) i jezične modele te potvrdilo rezultate. Provedena preliminarna istraživanja potvrdila su, a numerički rezultati prikazani tablicom *21* u poglavlju *6.4. Rezultati eksperimenata prvog ciklusa detekcije sličnosti na razini dokumenta* potvrđuju da je kosinusna sličnost pravi izbor mjere sličnosti potrebne za uparivanje s jezičnim modelima koji daju vektorske reprezentacije tekstova. U korist njezina korištenja govore i usporedivi rezultati različitih mjera sličnosti i udaljenosti (*Tablica 21. Prosječna uspješnost 60 metoda na pet korpusa (sortirano prema F1 mjeri)*), brzine izračuna (o čemu govori *Tablica 16. Vrijeme izvršavanja metoda na podskupu train korpusa Webis-11*) te izostanak potrebe preračunavanja rezultata mjera udaljenosti u rezultate sličnosti. Ona je ujedno i standardna mjera sličnosti vektora koja se koristi u NLP-u i mnogim drugim domenama tako da su ovakvi rezultati bili očekivani.

Za mjere sličnosti implementirane su kosinusna i meka kosinusna sličnost, a kao mjere udaljenosti korištene su: *Manhattan*, *Euclidean* i *Word Mover's Distance*. Svaka mjera daje određeni numerički rezultat iz intervala [-1, 1]. Taj se rezultat pretvara u binarni oblik {0, 1} koji klasificira rezultat kao sličan ili ne.

Za tu pretvorbu, binarizaciju, potrebno je odrediti graničnu vrijednost – prag binarizacije θ^*. Da bi se utvrdila njegova optimalna vrijednost, potrebno je decimalne rezultate metode binarizirati sa svim mogućim/smislenim vrijednostima praga binarizacije iz

intervala [0, 1], s korakom od 0.01. Dakle, svaka se vrijednost metode za svaki *train* podatkovni podskup svakoga korpusa provodi kroz petlju, rezultati se bilježe, određuje se maksimalna vrijednost F1-mjere za neku metodu i pripadni prag binarizacije kod kojega je došlo do toga maksimuma. Potom se na *test* podatkovnom podskupu koristi prethodno utvrđen optimalan prag binarizacije svojstven za svaku metodu kako bi se dobili konačni rezultati metode i rangirali prema F1-mjeri.

Sličnosti su izračunate za svaki par dokumenata, a dobiveni su rezultati uspoređeni sa službenim rezultatima iz označenih korpusa. U korpusu Clough & Stevenson, službene oznake nisu bile binarne. Zbog metodološke ujednačenosti s drugim standardnim korpusima (MSRP, Webis i P4PIN) te nebinarne oznake transformirane su u binarni oblik na način da su sve oznake koje upućuju na istovjetnost ili neki stupanj parafraziranja prevedene u 1, a ostale u 0. Točnost, preciznost, odziv, F1-mjera, MCC i matrica zabune računati su korištenjem *Pythonovih* modula *scikit-learn* (scikit-learn developers, 2021).

Proces izračuna rezultata pojedine metode i njezine evaluacije ponavljan je sa svim predviđenim metodama. Nakon što je utvrđena najbolja metoda, a s obzirom na to da je najuspješnija metoda ukazivala na potencijal još boljih rezultata iz obitelji modela koji daju najbolje rezultate, proveden je dodatni, drugi ciklus prve faze istraživanja, tijekom koje se na isti način evaluiralo 146 (velikih) jezičnih modela.

S obzirom na to da je u prvom ciklusu prve faze najuspješniji model bio inačica modela BERT i arhitekture transformera (prema rezultatima opisanima u poglavlju *6. Rezultati*), u **drugom ciklusu prve faze** istraživanja kombinirano je još 145 prethodno istreniranih jezičnih modela temeljenih na arhitekturi transformera uz korištenje različitih mjera sličnosti za izračun koeficijenta sličnosti između vektorskih reprezentacija tekstova dobivenih iz jezičnih modela.

Tijekom prve faze kroz eksperimente je korišteno pet označenih korpusa (CS, MSRP, Webis, P4PIN, VMEN). Četiri od njih javno su dostupna, dok je novi VMEN korpus načinjen za ovo istraživanje.

5.2.3. Pregled hiperparametara

Određivanje mnoštva hiperparametara programskog sustava koji daje najbolje rezultate, pri čemu veći dio tih parametara utječe na mnogobrojne modalitete obrade teksta, zahtijevalo je pripremu, programiranje i izvršenje velikog broja eksperimenata. Vrijednosti hiperparametara utvrđene su prema doprinosu boljim rezultatima mjerenima mjerom F1. Rezultati su potvrđeni Matthewsovim koeficijentom korelacije, korištenjem preliminarnih eksperimenata na jednostavnom testnom korpusu 10docs. Svaki je hiperparametar podešen nakon opsežnih eksperimenata i analize rezultata, s primarnim ciljem što boljeg utjecaja na rezultate, a sa sekundarnim ciljem ubrzanja obrade ili rješavanja tehničkih pitanja, tj. optimizacije procesa obrade na računalnim sustavima ograničenih mogućnosti. U tablici 12 nazivi hiperparametara poredani su abecednim redom, a uz njih podaci o korištenim tipovima podataka, podrazumijevajućim vrijednostima i kratkim opisima.

Tablica 12. Hiperparametri

Hiperparametar	Tip i vrijednost	Opis
alphanumeric	bool False	korištenje samo alfanumeričkih znakova
animation	bool True	stvaranje mp4 i gif animacija
backup	bool True	stvaranje arhiva
bigrams	bool False	rad s bigramima
clear_old_files	bool True	brisanje svih datoteka rezultata
corpus_file	str workingcorpus/my_corpus.txt	putanja datoteke s tekstom cijelog korpusa
dist2sim	str c	a, b, c, ab, ac, bc, abc; izračun sličnosti prema 1-3 formule
gpu_memory	int 3072	ograničenje korištenja GPU RAM-a
graph_calculate	bool True	izračun i crtanje grafičkih prikaza
graph_max_units	int 1000	maksimalan broj jedinica za prikaz na grafikonima
graph_show	bool False	prikaz grafikona na zaslonu
gwt	int 3	broj riječi koje se podudaraju u Greedy Word Tiling metodi
heatmap	bool False	stvaranje toplinske karte
histogram	bool False	stvaranje histograma

interactive	bool True	interaktivni (ili predefinirani) rad
iteration	int 10	broj iteracija kod treniranja jezičnih modela
learning_rate	float 0.0001	stopa učenja
mail	bool True	slanje email obavijesti o završetku rada
main_title	str Texts_Similarity_Recognition	glavni naslov
min_count	int 1	minimalna relevantna frekvencija riječi u korpusu
modeldl_load	bool True	korištenje prethodno pohranjenog jezičnog modela
modeldl_save	bool True	pohrana treniranog jezičnog modela
model_size	int auto	broj dimenzija modela
numbers	str None	tokens / delete / None = "keep" brisanje, čuvanje ili zamjena brojeva tokenima
pic_format	str png	izlazni format generiranih slika
post	bool False	označavanje riječi vrstom riječi (engl. *Part-of-speech tagging,* POS tagging)
ray	bool False	ray paralelizacija (Anyscale, 2024)
rounding	int 5	broj decimala kod zaokruživanja
scattergraph	bool False	stvaranje raspršenih grafikona
sigma	float 2	iznos sigme
sigma_activator	int 1000	minimalan broj riječi za aktivaciju sigme
single_letter_words	bool True	korištenje ili odbacivanje jednoslovnih riječi
simdump	str results	podmapa rezultata
small_letters	bool False	pretvorba u teksta u mala slova
sound	bool True	reproduciranje zvuka obavijesti
stemmer	str None	algoritmi za korjenovanje riječi (Snowball, Porter, Porter2 ili None)
stop_words	str keep	brisanje, čuvanje ili zamjena stop riječi
table_size	int 10	broj prikazanih i spremljenih stupaca/redova tablica rezultata
top_freq_words	int 500	broj najfrekventnijih riječi u grafičkom prikazu
transparent	bool True	grafički prikazi s prozirnom pozadinom
vector_combine	str mean	kombiniranje vektora riječi u vektor rečenice (mean, sum = srednja vrijednost, zbroj)
vector_normalize	bool True	normalizacija vektora nakon kombiniranja
vector_projections	str None	vrsta projekcije vektora u 2D grafički prikaz: None, PCA (default), t-SNE, All
window	int 5	prozor riječi u korpusu (ispred i iza središnje)
wn_hnym	str None	korištenje hiperonima, holonima (hypernym, holonym or None)
wn_lemma	bool False	lematizacija

wn_morphy	bool False	morfološka analiza (implicira i lematizaciju)
path	str corpuses/10docs/	putanja do korpusa
jobs	list 19-25	brojevi i/ili intervali modula predviđeni za komandno-linijsko izvođenje

Svaki parametar ima najmanje dvije moguće vrijednosti, često i nekoliko njih, a podrazumijevajuće su vrijednosti utvrđene na temelju rezultata nastalih uzastopnim eksperimentima s izmjenama pojedinoga parametra.

5.2.4. Određivanje broja dimenzija vektorskog prostora

Broj parametara neuronske mreže korištenih kod treniranja nekog modela izravno utječe na njegove performanse te je jedan od čimbenika koji utječe na performanse LLM-ova, a preostali su: građa modela (Fedus i sur., 2022; Lepikhin i sur., 2020; Shen i sur., 2019), kvaliteta i veličina tekstnog ulaza za treniranje, broj dimenzija vektorskog prostora modela te broj hiperparametara modela. kod modela dubokog učenja iz prve faze istraživanja, dakle za Word2Vec, Doc2Vec, FastText i GloVe, jezični modeli potrebni za dobivanje vektorskih reprezentacija riječi i/ili cjelovitih tekstova, dobiveni su njihovim (vlastitim) treniranjem nad korpusima nad kojima se traži parafraziranje. U prvoj fazi istraživanja, kod treniranja navedenih modela[1], jedan od važnih parametara njihova učenja modela je broj dimenzija vektorskog prostora koji nastaje njegovim treniranjem. Logičko promišljanje navodi nas na zaključak da je većim brojem dimenzija moguće kvalitetnije vektorski reprezentirati tekst. Jednako tako, iz trenda rasta broja dimenzija jezičnih modela od prvih do današnjih velikih, očito da veći broj dimenzija utječe na bolju kvalitetu modela, odnosno na njegovu sposobnost predstavljanja semantike iz tekstova. Stoga je u okviru ovoga istraživanja bilo potrebno istražiti kako je broj dimenzija povezan s performansama modela. Prva naznaka da porast broja dimenzija ne vodi linearno do porasta kvalitete rezultata mjerenih F1-mjerom evidentirana je tijekom provedbe velikog broja eksperimenata na razvojnom korpusu 10docs. Njima je utvrđeno da nakon „prekomjernog" povećanja broja dimenzija, performanse modela zapravo – padaju. Empirijskom metodom pokušalo se stoga utvrditi varijable i funkcije nad njima kojima bi se mogao izračunati broj dimenzija modela koji je optimalan kod treniranja kako bi model prenio semantiku teksta kojeg preslikava u vektorski prostor. Optimalnost se odnosi na delikatnu ravnotežu minimalnog broja dimenzija kod koje model daje vrhunske rezultate. To nije broj dimenzija kod kojega model ostvaruje apsolutno najbolji rezultat već kod toga broja model ostvaruje rezultate dovoljno visoke, usporedive s maksimalnim mogućim za taj model. Istraživači se rijetko bave formalnim definiranjem funkcije koja bi dala optimalan broj dimenzija za proizvoljan korpus. Među rijetkima Luan i sur. (2021) predložili su vrijednost $\sqrt{max_d \|d\|}$ tj. drugi korijen od maksimalne dužine[2] najvećeg

1 Neki modeli imaju i svoje podmodele, pa tako Word2Vec ima CboW i Skip-Gram, a Doc2Vec ima DBoW i DM. Dodatno Doc2Vec i GloVe imaju izlaze (vektorske reprezentacije) kako za pojedine pojmove tako i za cjelokupni tekst.

2 Iz konteksta rada pa ni iz domene računalne lingvistike kojoj pripada, nije moguće razaznati smatraju li

dokumenta u korpusu (Luan i sur., 2021). Drugi je primjer ekstrakcija svojstava (engl. *features*) iz korpusa te predviđanje optimalne dimenzije korištenjem regresijskog modela *random forest* (Gao i sur., 2021). Autori istraživanja (Nalisnick i Ravi, 2017) predložili su pristup modifikacije modela Word2Vec tako da on nauči dimenzionalnost iz podatkovnog skupa – korpusa, i to na probabilistički način. Drugi istraživači, ako se i bave tim problemom, zadovoljavaju se pristupom kojim nakon treniranja modela reduciraju njegovu dimenzionalnost.

U obliku formule, prva dobra aproksimacija optimalnog broja dimenzija korištena u istraživanju Vrbanec i Meštrović (2021a) definirana je funkcijom

$$Dim(NW) = min(10 \cdot ceil(\log_2 NW), 300) \tag{46}$$

gdje je *NW* broj jedinstvenih riječi u korpusu, a *ceil* je matematička funkcija *najveće cijelo* koja zaokružuje decimalni broj na prvi sljedeći cijeli broj (Vrbanec i Meštrović, 2021a). Zahvaljujući mnogim kasnijim eksperimentima pronađena je jednostavnija i preciznija aproksimacija optimalnog broja dimenzija (Vrbanec i Meštrović, 2023) funkcijom

$$Dim(NW, ND) = log_2 NW \cdot log_2 ND \tag{47}$$

gdje su *NW* i *ND* broj jedinstvenih riječi u korpusu i broj (ne praznih) dokumenata odnosno tekstova u korpusu.

kod predtreniranih modela njihov broj dimenzija već je zadan od njihovih tvoraca, pa problematika optimalnog broja dimenzija za naše eksperimente pada u drugi plan, no ne i kao teorijski koncept koji bi bilo dobro koristiti kod treniranja budućih velikih jezičnih modela. Vidljivo je da su, gledajući broj dimenzija velikih jezičnih modela prikazanih u tablici 6, brojke prilično diskretne veličine, tj. istraživači koriste ograničeni skup brojeva za dimenzije svojih modela koji su iz skupa $\{256=2^8,\ 512=2^9,\ 768=2^8+2^9,\ 1024=2^{10},\ 1600=100\cdot2^4,\ 2048=2^{11},\ 4096=2^{12},\ 8192=2^{13},\ 12288=2^{12}+2^{13},\ 16384=2^{14},\ 20480=10\cdot2^{11}\}$. Stoga se nameće zaključak da su brojevi dimenzija tehnički motivirani računalnim sustavima kojima raspolažu i njihovim ograničenjima te optimiziranjem parametara treniranja ili kasnijega korištenja modela.

Kako bismo bili sigurni da je prirodna veza između broja dimenzija vektorskog prostora koji koriste jezični modeli logaritamske prirode i da bi se prethodno aproksimirane formule potvrdile, provedeni su dodatni eksperimenti čiji su rezultati prikazani tablicama 13 i

autori pod dužinom dokumenta dužinu stringa koji čini dokument ili broj riječi u dokumentu.

14. Tablice sadrže vrijednosti F1-mjere koje su prikazane kao rezultati ovisnosti o broju dimenzija modela na dva korpusa i dva jezična modela.

Tablica 13. Vrijednosti F1-mjere za različite dimenzije modela (korpus MSRP)

Broj dimenzija	F1-mjera MSRP (Doc2Vec DBoW)	F1-mjera MSRP (Word2Vec CBOW)
16	0.797	0.797
32	0.795	0.794
64	0.795	0.797
128	0.809	0.810
256	0.809	0.808
444	0.809	0.809
456	**0.812**	**0.813**
511	0.798	0.798
512	0.814	0.814
513	0.806	0.805
620	0.801	0.805
768	0.805	0.806
949	0.807	0.807
1024	0.800	0.807
2048	**0.817**	**0.810**
4096	0.815	0.816

Crvenom su bojom označene vrijednosti za formulom (41) izračunat optimalan broj dimenzija, dok su podebljane najbolje vrijednosti za 16 različitih dimenzija odabranih u istraživanju na način da su ili potencije baze dva ili dimenzije izračunate po nekoj od formula, ili jednostavno dimenzija koja je u određenome smislu zanimljiva ili značajna (s ciljem utvrđivanja rastućega ili padajućega trenda, tj. lokalnog maksimuma).

Tablica 14. Vrijednosti F1-mjere za različite dimenzije modela (korpus P4PIN)

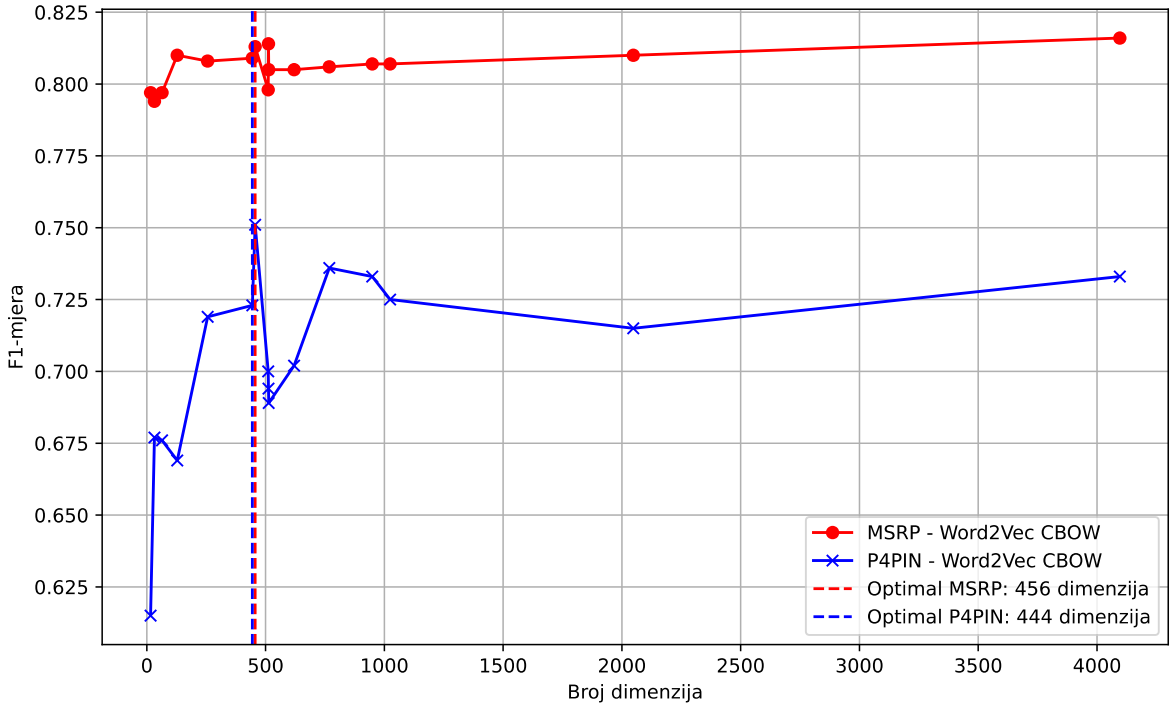
Broj dimenzija	F1-mjera P4PIN (Doc2Vec DBoW)	F1-mjera P4PIN (Word2Vec CBOW)
16	0.606	0.615
32	0.680	0.677
64	0.675	0.676
128	0.735	0.669
256	0.718	0.719
444	**0.709**	**0.723**
456	**0.750**	**0.751**
511	0.705	0.700
512	0.721	0.694
513	0.727	0.689
620	0.725	0.702
768	0.732	0.736

Broj dimenzija	F1-mjera P4PIN (Doc2Vec DBoW)	F1-mjera P4PIN (Word2Vec CBOW)
949	0.731	0.733
1024	0.737	0.725
2048	0.724	0.715
4096	0.727	0.733

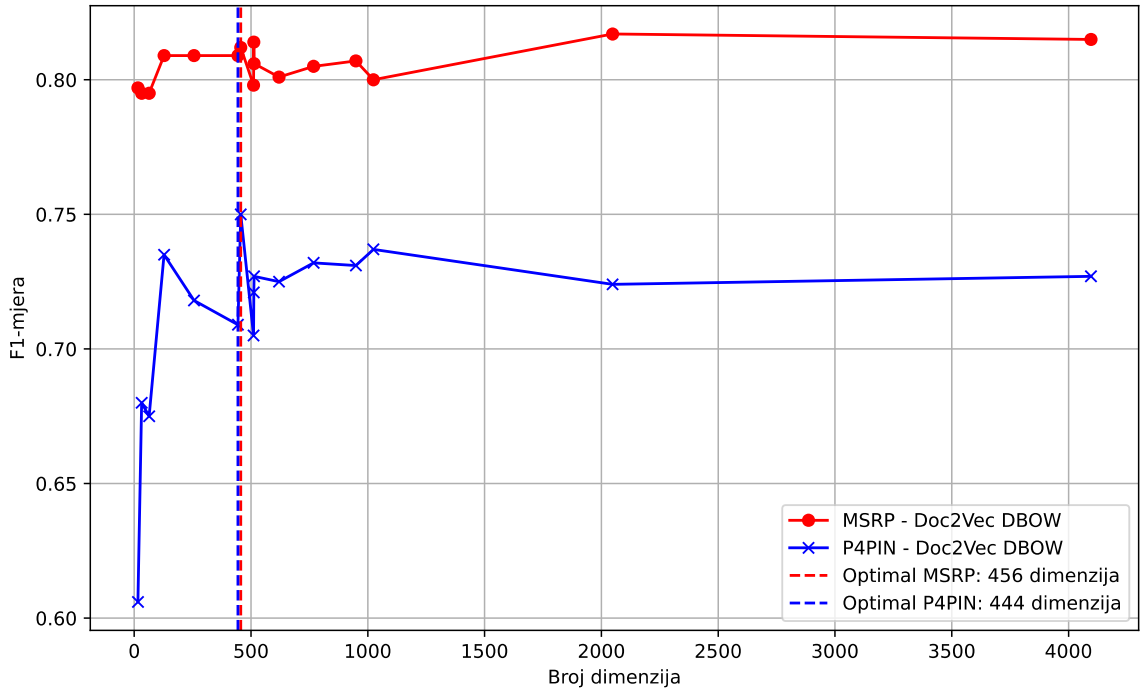
Brojevi iz tablica 13 i 14 grafički su prikazani slikama 20-22 koje ukazuju da je funkcija koja opisuje ovisnost performansi modela od broja njegovih dimenzija doista logaritamske prirode. Štoviše, na temelju broja jedinstvenih riječi i broja dokumenata dvaju korpusa te rezultata eksperimenata, formula za približno optimalan broj dimenzija modificirana je te konačno definirana funkcijom

$$Dim(NW, ND) = 4 \cdot log_2 NW \cdot log_2 ND \tag{48}$$

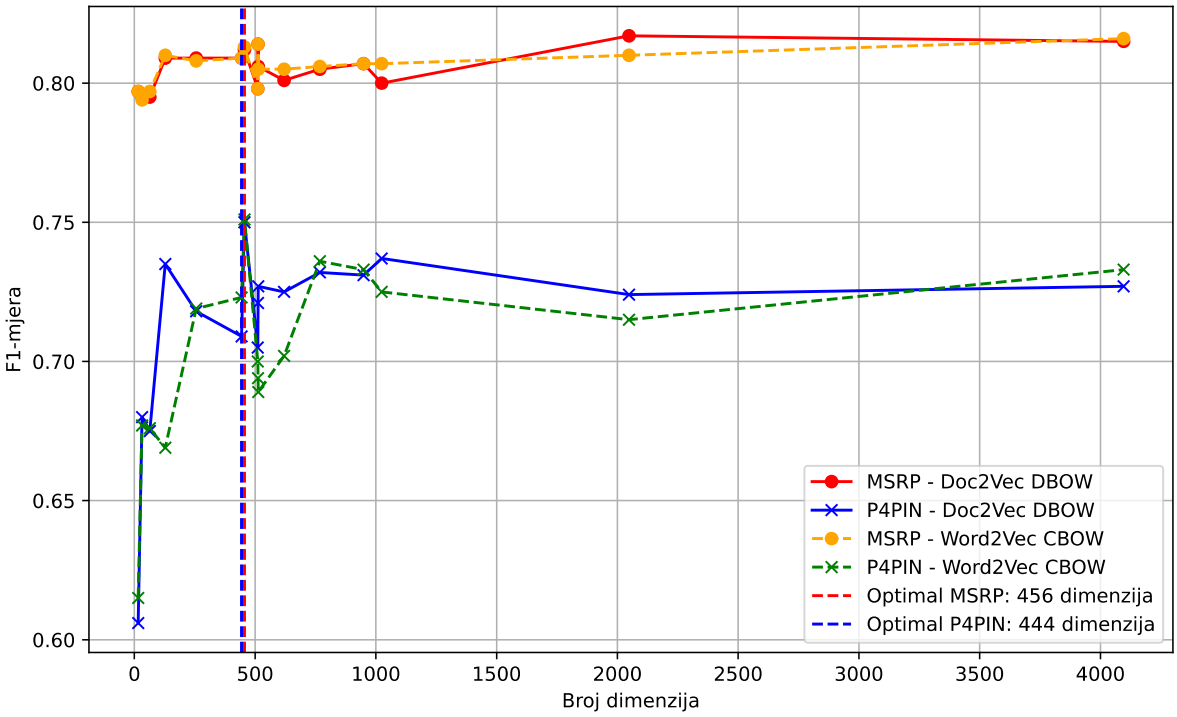
gdje su *NW* i *ND* broj jedinstvenih riječi u korpusu i broj (ne praznih) dokumenata / tekstova u korpusu. Nova formula uključuje faktor 4. Naime, prema rezultatima eksperimenata, kod izračuna približno optimalnoga broja dimenzija, vrijednosti *NW* i *ND* potrebno je kvadrirati, a potom ti kvadrati kao eksponenti argumenata binarnog logaritma izlaze ispred umnoška logaritama kao dodatni koeficijent 4 (formula *48*).



Slika 20. Odnos broja dimenzija i F1 vrijednosti (Word2Vec CBOW) – 16 vrijednosti



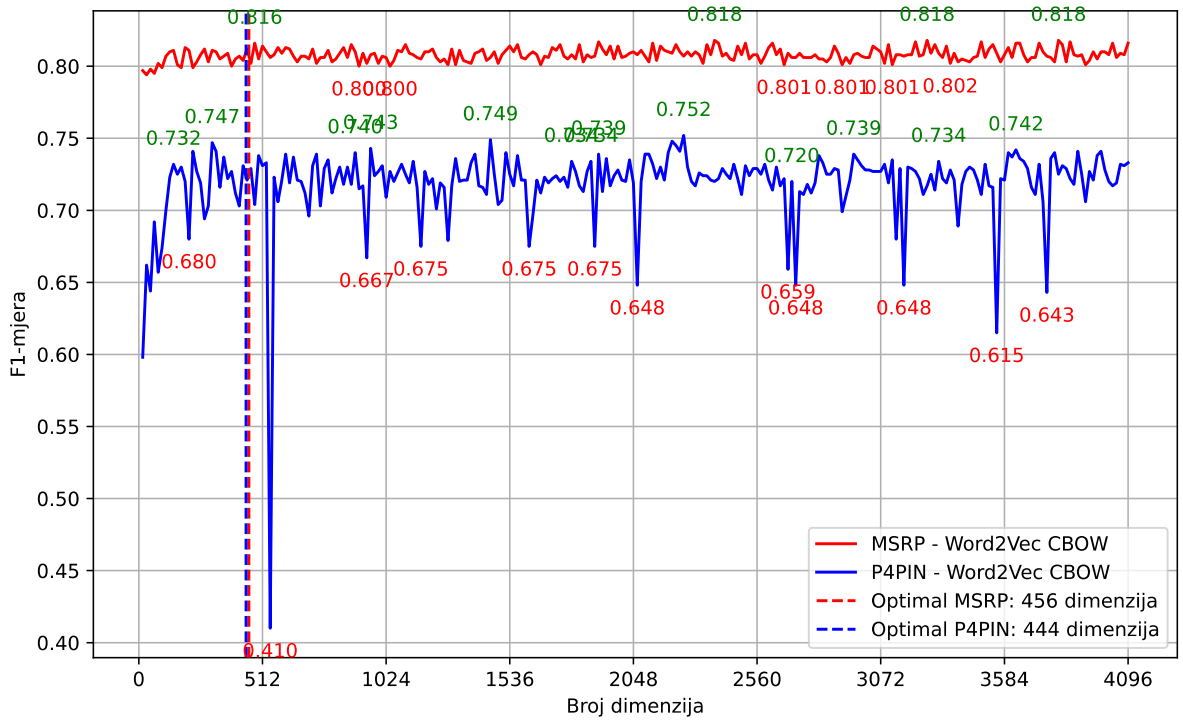
Slika 21. Odnos broja dimenzija i F1 vrijednosti (Doc2Vec DBoW) – 16 vrijednosti



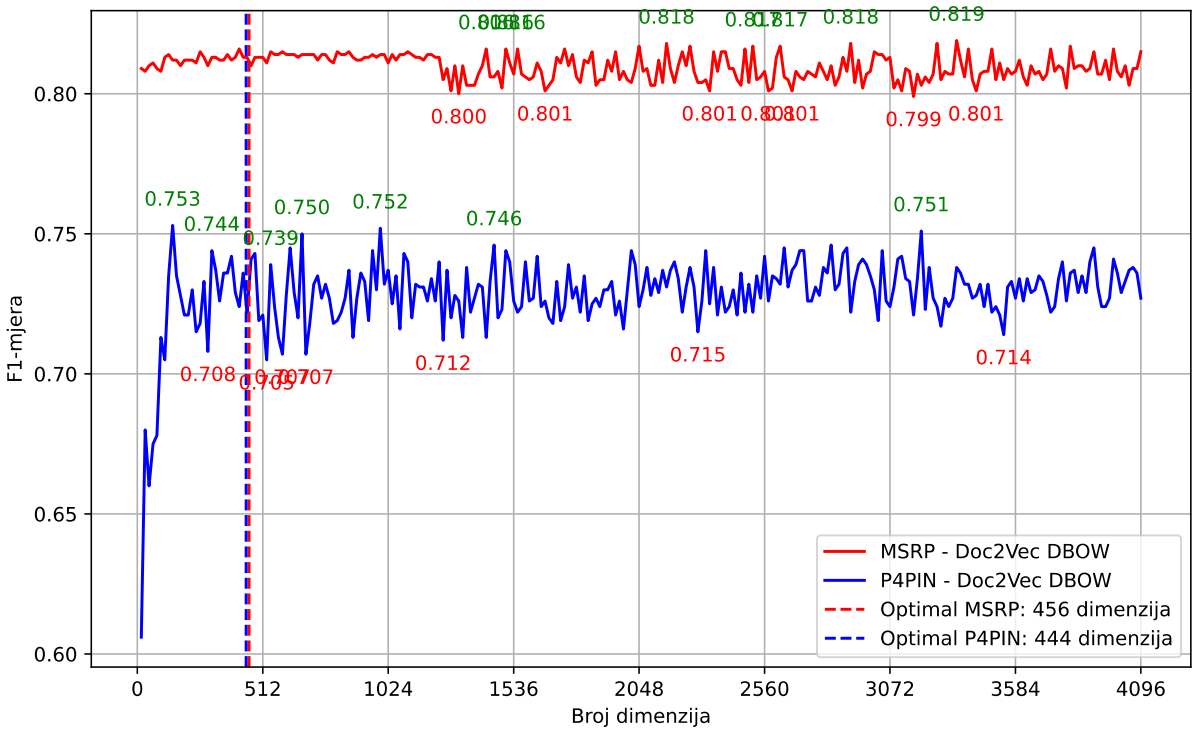
Slika 22. Usporedni prikaz odnosa broja dimenzija i F1 vrijednosti – 16 vrijednosti

Iako globalno prate logaritamsku funkciju, grafovi na slikama 20-22, koji prikazuju vrijednosti iz provedenih eksperimenata, imaju anomalije, tj. grafovi nisu glatke linije. Primjerice, za MSRP korpus oko 2^9=512, dakle na broju dimenzija 511 i 513 vrijednosti F1-mjere su nešto niže od vrijednosti u 512, tj. 512 predstavlja lokalni maksimum. Pojava lokalnih maksimuma vidljiva je za oba jezična modela, ali za različite korpuse na različitim mjestima i različitih amplituda oscilacija.

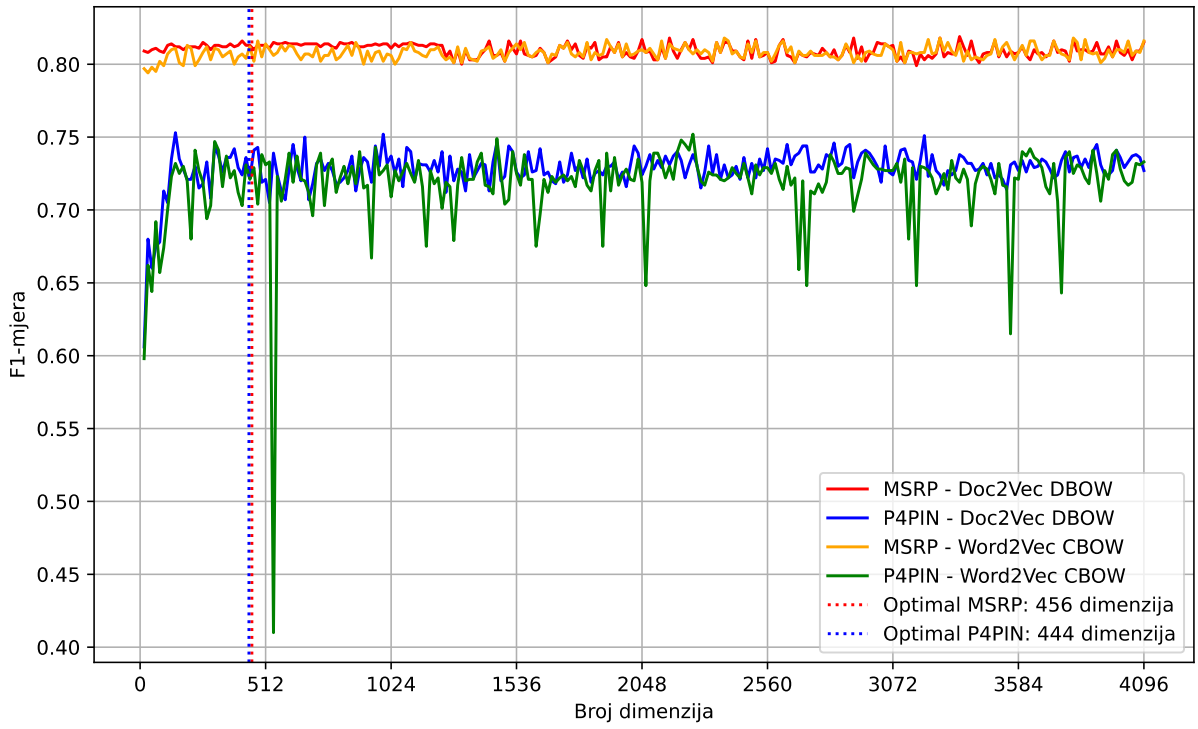
Da bi se još bolje vidjela priroda funkcije koja opisuje kretanje adekvatnosti vektorskih reprezentacija u odnosu na broj dimenzija korištenoga vektorskog prostora, a posebice zbog uočenih anomalija, u novoj seriji eksperimenata korišteno je 256 vrijednosti dimenzija od 16 do 4096, s korakom od 16, tj izračunate su vrijednosti F1-mjere za dva korpusa i dva modela s obzirom na dimenzije vektorskih reprezentacija. Rezultati su prikazani u tablici 39, u poglavlju *Dodatak: Vrijednosti F1-mjere s obzirom na dimenzije vektorskih reprezentacija* te u nastavku, slikama 23-25, ovoga puta i s uključenim vrijednostima lokalnih maksimuma i minimuma.



Slika 23. Odnos broja dimenzija i F1 vrijednosti (Word2Vec CBOW) – 256 vrijednosti



Slika 24. Odnos broja dimenzija i F1 vrijednosti (Doc2Vec DBoW) – 256 vrijednosti



Slika 25. Usporedni prikaz odnosa broja dimenzija i F1 vrijednosti – 256 vrijednosti

Lokalne ekstremne vrijednosti F1-mjere ostaju nerazjašnjene, a logično objašnjenje može biti u implementacijama algoritama u *Python gensim* modulu.

Najvrjednija je spoznaja prethodnih grafikona ta da je očekivana vrijednost F1-mjere izračunate kod optimalnog broja dimenzija blizu globalne maksimalne vrijednosti te je povećanje broja dimenzija nepotreban „trošak" računalnih resursa kod treniranja, što se pak odražava na vremenski i financijski trošak treniranja. Formula (48) koja daje optimalan broj dimenzija vektorskog prostora ovisno o dva parametra: broj jedinstvenih riječi u korpusu te broj nepraznih dokumenata u korpusu, ima potencijal ostvariti značajne uštede u pogledu trajanja treniranja i za to potrebnih računalnih resursa, posebno kod treniranja velikih jezičnih modela koje traje mjesecima na iznimno moćnim računalnim klasterima.

5.2.5. Konstrukcija vektorske reprezentacije teksta

Neki jezični modeli (Word2Vec, FastText, GloVe), iz prve faze istraživanja sa 60 metoda, prije korištenja predtreniranih modela, ne generiraju vektorsku reprezentaciju dokumenata ili većih tekstnih jedinica već samo vektorske prezentacije riječi. Za te je modele bilo potrebno kreirati vektorske reprezentacije dokumenata iz vektorskih reprezentacija riječi, npr. u oblicima:

- nenormalizirani centroid (srednja vrijednost nenormaliziranih vektora riječi)

$$V_{nnc} = \frac{\sum_{v \in V} v}{|V|} \tag{49}$$

kao što je opisano u (Babić i sur., 2019, 2020);

- normalizirani centroid (normalizirana srednja vrijednost nenormaliziranih vektora riječi)

$$V_{nc} = \frac{\sum_{v \in V} v}{|V| \cdot \left| \sum_{v \in V} v \right|} \quad ; \tag{50}$$

- normalizirana srednja vrijednost vektora riječi gdje se svaki vektor riječi prvo normalizira, a potom se oni zbroje.

$$V_{nsv} = \frac{\sum_{v \in V} \frac{v}{|v|}}{|V|} \quad . \tag{51}$$

S obzirom na to da ni jedna postojeća metoda kombiniranja vektora riječi u vektore

dokumenata nije prihvaćena kao najbolja i da je isprobano nekoliko metoda kombiniranja, odabrana je ona koja je donijela najbolje rezultate, a to je normalizirana srednja vrijednost nenormaliziranih vektora riječi koji čine tekstove.

5.3. Eksperimenti za utvrđivanje parafraziranja na razini rečenica

Eksperimenti nad rečenicama kao temeljnim misaonim jedinicama teksta važni su za provedbu treće faze metode DLPDM, a imaju za cilj identifikaciju parafraziranja na razini rečenica. Dakle, u trećoj su fazi između dovoljno sličnih dokumenata uspoređivane rečenice, u potrazi za parafraziranjima. Analizom lažno pozitivnih i lažno negativnih slučajeva postalo je jasno da na konačni rezultat mjerenja parafraziranosti utječe više parametara, tj. da iako rezultati mjere kosinusne sličnosti nad vektorima jezičnih modela temeljenih na dubokom učenju (kraće: mjera sličnosti modela) daju jako dobre rezultate, bolje od do sada najboljih, postoji prostor za dodatno poboljšanje.

Empirijskim postupkom, računanjem različitih mogućih kombinacija mjera sličnosti i parafraziranosti, ono je i postignuto na način da je korištena kvadratna sredina s tri ponderirane mjere sličnosti (mjera semantičke sličnosti modela, mjera GWT i mjera sličnosti dužine riječi). Navedene tri mjere, različito ponderirane, unose se u kvadratnu sredinu koja za ulazne vrijednosti iz intervala [0, 1] daje vrijednosti manje ili jednake od aritmetičke sredine, ali naglašava utjecaj većih vrijednosti. Mjera DLCPM implementirana je u programskom jeziku *Python* na temelju formula definirane u poglavlju *4.2.3. Istraživanje postupaka detekcije parafraziranja na razini rečenica*.

Treba napomenuti da je za potrebe definiranja nove mjere sličnosti u preliminarnoj fazi istraživanja proveden niz eksperimenata koji su uključivali različite varijacije mjera sličnosti. Prema tim rezultatima preliminarnih eksperimenata, navedena kombinacija mjera i sredine, nova DLCPM mjera, rezultirala je poboljšanim rezultatima u odnosu na one dobivene korištenjem samo jedne mjere sličnosti.

Treća faza metode DLPDM provodi se samo između onih parova dokumenata kojima je u prethodnoj fazi utvrđena sličnost veća od zadanoga praga. Dakle, s modelom koji se pokazao najboljim odabirom za utvrđivanje sličnosti na razini dokumenta nastavilo se u drugu fazu – usporedbe dokumenata na rečeničnoj razini kako bi se detektiralo parafraziranje. U toj fazi istraživanja usporedba je izvršena na razini rečenica koje su ekstrahirane iz dokumenata

korištenjem specifičnih regularnih izraza. Ti izrazi pokazali su se učinkovitijima i bržima u odnosu na korištenje biblioteka *nltk* i *spacy*.

S obzirom na to da je za istraživanje parafraziranosti parova rečenica potrebno imati oznake na rečeničnoj razini, bilo je moguće koristiti samo tri korpusa (MSRP, P4PIN i VMENAIA) jer su otpali korpusi koji nemaju te oznake. Strogo gledajući ni VMENAIA korpus koji je modificirana, tj. unaprijeđena verzija VMEN korpusa, nema oznake na razini rečenica, ali upravo je njegova modifikacija (VMEN => VMENAIA) omogućila oznake i na rečeničnoj razini (vidi *4.3.5. Korpusi VMEN i VMENAIA*).

U okviru provedenih eksperimenata utvrđen je optimalni prag za određivanje parafraziranja, θ^{**}. Ako DLCPM mjera sličnosti para rečenica premaši utvrđeni prag, ispitivana se rečenica u odnosu na uspoređivanu proglašava parafraziranom. Znajući ukupan broj rečenica provjeravanog dokumenta i broj rečenica koje su parafrazirane, moguće je na kraju izračunati i postotak parafraziranosti provjeravanog dokumenta u cjelini i u odnosu prema uspoređenim dokumentima.

Rezultati na temelju evaluacije pomoću F1-mjere i MCC-a prikazani su u poglavlju *6. Rezultati*.

5.4. Analiza složenosti

U prvom dijelu istraživanja uspoređeno je 60 različitih metoda izračuna sličnosti dokumenata/tekstova koristeći pet različitih korpusa parafraziranih tekstova. Međutim rezultati F1 i MCC-a nisu jedini parametri koji definiraju prikladnost metoda za buduću upotrebu. Jedan je parametar vrlo pragmatičan: odabrana metoda trebala bi biti brza i što manje zahtjevna prema računalnim resursima.

5.4.1. Složenost u *O* notaciji

Jedan od načina za procjenu metoda prema tim kriterijima je korištenje *O* notacije za opisivanje algoritamske, ujedno i vremenske složenosti algoritma/metode. Predstavlja gornju granicu broja operacija koje će algoritam/metoda izvesti kako veličina ulaza raste (s obzirom na svoj algoritam te broj tekstova/dokumenata i broj riječi korištenih u korpusima). Navedene procjene složenosti predstavljene su u tablici 15. Složenosti predstavljene u tablici s *O* (*N*, *M*, *T*, *I*, *K*, *D*) zapisom definirane su veličinom korpusa – brojem dokumenata (*N*), brojem

jedinstvenih pojmova u korpusu (*M*), brojem korištenih tema (*T*), brojem iteracija koje se koriste za neke modele (*I*), brojem dimenzija koje se koriste za predstavljanje svakog dokumenta (*K*) i brojem dimenzija vektorskih reprezentacija (*D*). Svaka procjena složenosti izvršena je analizom programskoga koda koji je korišten u eksperimentima, u interakciji s ChatGPT generativnim jezičnim modelom kome je predan programski kod za svaku metodu, a dobiveni je rezultat (izraz složenosti u *O* notaciji u obliku algebarskog izraza) logički i matematički ispravljen (zbog halucinacija te uočavanja elegantnijih i matematički jednostavnijih izraza), unificiran (tako da se za iste parametre *O* notacije uvijek koriste iste oznake) te minimiziran klasičnim matematičkim alatom minimizacije algebarskih izraza.

Tablica 15. Složenost korištenih metoda izražena *O* notacijom

Metoda	***O*(*N*, *M*, *T*, *I*, *K*, *D*)**
tfidf	$O(N^2 \cdot M \cdot logN)$
lsi	$O(N(M^2+N \cdot K))$
lda	$O(T \cdot N \cdot M \cdot logM)$
hdp	$O(N \cdot I \cdot logM)$
rp	$O(N \cdot M^2)$
le	$O(N \cdot M^2)$
jccrd	$O(N^2)$
lev	$O(N^2)$
gwt	$O(N^2 \cdot M \cdot logM)$
cosw2vcbow	$O(N^2 \cdot M)$
cosw2vsg	$O(N^2 \cdot M)$
cosd2vwdbow	$O(N(M \cdot K \cdot I \cdot logI+N \cdot K))$
cosd2vwdm	$O(N(M \cdot K \cdot I \cdot logI+N \cdot K))$
cosd2vddbow	$O(N(M \cdot K \cdot I \cdot logI+N \cdot K))$
cosd2vddm	$O(N(M \cdot K \cdot I \cdot logI+N \cdot K))$
cosft	$O(N(M+K \cdot logK+N))$
cosglw	$O(N^2(M+K))$
cosgld	$O(N^2(M+K))$
cosuse	$O(N^2)$
coselmo	$O(N(M+N \cdot K))$
cosbert	$O(N^2 \cdot K)$
coslaser	$O(N^2 \cdot M)$
edw2vcbow	$O(N^2 \cdot D)$
edw2vsg	$O(N^2 \cdot D)$
edd2vwdbow	$O(N(N \cdot D+M))$
edd2vwdm	$O(N(N \cdot D+M))$
edd2vddbow	$O(N(N \cdot D+M))$
edd2vddm	$O(N(N \cdot D+M))$
edft	$O(N^2 \cdot D)$
edglw	$O(N^2 \cdot D)$

Metoda	***O*(*N*, *M*, *T*, *I*, *K*, *D*)**
edgld	$O(N^2 \cdot D)$
eduse	$O((D \cdot N)^2)$
edelmo	$O((D \cdot N)^2)$
edbert	$O(M \cdot N \cdot D^2)$
edlaser	$O(N^2)$
mdw2vcbow	$O(N^2 \cdot K)$
mdw2vsg	$O(N^2 \cdot K)$
mdd2vwdbow	$O((D \cdot N)^2)$
mdd2vwdm	$O(M \cdot N \cdot D^2)$
mdd2vddbow	$O(N^2 \cdot D)$
mdd2vddm	$O(M \cdot N \cdot D^2)$
mdft	$O(N^2 \cdot D)$
mdglw	$O(N(M+N \cdot K+N))$
mdgld	$O(N^2 \cdot D)$
mduse	$O(N^2 \cdot D)$
mdelmo	$O(N(D^2+N))$
mdbert	$O(N^2 \cdot D)$
mdlaser	$O(N^2 \cdot D)$
scsw2vcbow	$O(N \cdot D(M+logD))$
scsw2vsg	$O(N(K \cdot logK+M \cdot logD+D \cdot logN))$
scsd2vwdbow	$O(N \cdot D(I+M+logD+N))$
scsd2vwdm	$O(T \cdot I \cdot N^2(D+M \cdot logM))$
scsft	$O(N^2 \cdot K)$
scsglw	$O(N^2(M+ND))$
wmdw2vcbow	$O(N^2 \cdot M \cdot logM)$
wmdw2vsg	$O(N(M+K \cdot I \cdot logI+NK))$
wmdd2vwdbow	$O(N(M+logD+K \cdot logI+I^2+N))$
wmdd2vwdm	$O(N \cdot K(M+N \cdot I \cdot logK+N))$
wmdft	$O(N(M \cdot logM+N \cdot I \cdot logI+N \cdot logM))$
wmdglw	$O(N \cdot K(M+N \cdot logK+N))$

Popis iz prethodne tablice nije sortiran prema složenosti. Za to bi bila potrebna dodatna analiza, uz pretpostavljene maksimalne vrijednosti svih parametara, što i dalje ne bi jamčilo ispravno rangiranje. Grubo gledajući, ako kao parametar usporedbe uzmemo zbroj eksponenata koje pojedine veličine imaju, vidljivo je da metoda *cosbert* koja je u ukupnom poretku svih metoda nad pet korpusa bila najuspješnija mjereći uspješnost mjerom F1, stoji vrlo dobro u pogledu složenosti predstavljene procijenjenom *O* notacijom.

5.4.2. Vrijeme izvršavanja kao odraz složenosti

Drugi način za mjerenje prikladnosti metoda mjerenja sličnosti tekstova za daljnju upotrebu u otkrivanju parafraziranja jest mjerenje i bilježenje vremena koje računalo utroši za

izračun metoda. Time se vremenska potrošnja koristi kao pokazatelj performansi metode. Rezultati su prikazani tablicom 16. Golem je problem predstavljala dugotrajnost izvršavanja programa *Python* za neke metode i mjere udaljenosti, osobito GWT i WMD na velikim korpusima, a posebno na najopsežnijem korpusu (Webis-11). Tijekom razvoja programa trebalo je savladati mnogobrojne probleme koji proizlaze iz složenosti metoda. Ti su problemi bili povezani s potrošnjom radne memorije i paralelizmom izvršavanja (višedretvenošću). Cilj programiranja provedbe svake metode bio je isti: koristiti sve dostupne jezgre u kritičnim dijelovima metode i maksimalno učinkovito korištenje RAM-a. Minimiziranje upotrebe virtualne memorije bilo je presudno jer vrijeme potrebno za izračune eksponencijalno raste kada se ona koristi. Tablica 16 prikazuje vremena potrebna za izvršavanje metoda, odnosno utrošeno vrijeme da metoda izračuna matricu sličnosti između dokumenata/tekstova za podskup tekstova *train* najvećega korpusa (Webis-11). Kratice i njihova značenja objašnjeni su u poglavlju *Dodatak: Kratice*.

Tablica 16. Vrijeme izvršavanja metoda na podskupu *train* korpusa Webis-11

#	Model	Vrijeme[s]	Vrijeme[h]
1	cosw2vcbow	6.6	0.002
2	cosw2vsg	14.3	0.004
3	le	19.4	0.005
4	tfidf	25.7	0.007
5	cosd2vwdm	29.4	0.008
6	cosd2vwdbow	30.9	0.009
7	cosgld	32.9	0.009
8	scsd2vwdbow	42.9	0.012
9	cosd2vddbow	49.4	0.014
10	scsd2vwdm	49.4	0.014
11	cosd2vddm	54.6	0.015
12	cosft	60.7	0.017
13	lsi	65.3	0.018
14	scsw2vcbow	77.3	0.021
15	hdp	80.9	0.022
16	**cosbert**	**86.4**	**0.024**
17	cosglw	132.1	0.037
18	scsw2vsg	220.6	0.061
19	rp	274.8	0.076
20	lda	290.7	0.080
21	coslaser	849.3	0.236
22	cosuse	854.9	0.237
23	jacc	873.5	0.243
24	mdd2vwdm	2543.0	0.706

#	Model	Vrijeme[s]	Vrijeme[h]
25	mdd2vwdbow	2559.8	0.711
26	mdw2vcbow	2570.8	0.714
27	mdft	2589.9	0.719
28	edw2vcbow	2600.5	0.722
29	edd2vwdm	2619.0	0.728
30	edd2vwdbow	2628.7	0.730
31	mdw2vsg	2669.1	0.741
32	edglw	2693.9	0.748
33	mdglw	2696.5	0.749
34	edft	2701.8	0.751
35	edw2vsg	2755.9	0.766
36	edd2vddbow	2994.9	0.832
37	mdgld	3005.5	0.835
38	edd2vddm	3030.6	0.842
39	mdd2vddm	3030.8	0.842
40	mdd2vddbow	3035.1	0.843
41	edgld	3040.2	0.845
42	edbert	3708.9	1.030
43	mdbert	3781.2	1.050
44	mdlaser	3831.2	1.064
45	mduse	4381.9	1.217
46	eduse	4450.7	1.236
47	edlaser	4667.5	1.297
48	coselmo	5124.8	1.424
49	edelmo	8816.6	2.449
50	mdelmo	9056.2	2.516
51	scsft	13539.3	3.761
52	scsglw	13539.3	3.761
53	lev	27221.3	7.561
54	wmdglw	59558.3	16.54
55	wmdw2vcbow	62911.0	17.48
56	wmdft	64328.0	17.87
57	wmdw2vsg	230523.9	64.03
58	wmdd2vwdm	237617.5	66.01
59	wmdd2vwdbow	349185.9	97.00
60	gwt	709079.9	197.0

Vrijednosti vremena koje su bile potrebne metodama i koje su prikazane u tablici 16 dovode do zaključka da je upotreba mjere WMD udaljenosti neprikladna zbog vrlo velikog utroška vremena, višestruko većega od ostalih mjera. GWT je još i više zahtjevna prema računalnim resursima od WMD-a, iako je utrošeno jako mnogo vremena korištenjem svake moguće optimizacijske tehnike i heuristike kako bi postalo moguće dovršavanje izračuna u vremenu od 197 sati. No, od GWT kao metode nije se moglo odustati jer nam je ona bila

nužna kao jedna od ključnih osnovnih vrijednosti (engl. *baseline*), za usporedbu s drugim metodama, posebno onima temeljenim na DL-u. Metoda *cosbert* (podebljane vrijednosti u tablici 16), koja koristi jezični model BERT odnosno njegov vektorski prostor i kosinusnu sličnost između vektora te koja je najbolje rangirana prema rezultatima F1 (vidi poglavlje *6.4. Rezultati eksperimenata prvog ciklusa detekcije sličnosti na razini dokumenta*), nije najučinkovitija u vremenskoj potrošnji, ali je i u tom pogledu ostvarila vrlo dobar rezultat te opravdala uporabu u sljedećim fazama istraživanja smjestivši se na samu granicu skupine metoda s brzim izračunom.

Reimers i Gurevych (2019) primijetili su da kada se traži semantička sličnost između dviju rečenica, modeli BERT obitelji zahtijevaju da se obje rečenice unesu u mrežu, što uzrokuje golemo računalno opterećenje: pronalaženje najsličnijeg para u skupu od 10000 rečenica zahtijeva oko 50 milijuna inferencijskih izračuna (~65 sati). Prema istim autorima, konstruiranje nekih modela obitelji BERT čini ih neprikladnima za izračunavanje semantičke sličnosti (Reimers i Gurevych, 2019). Suprotno tim iskustvima Reimersa i Gurevycha, korištenje unaprijed treniranih modela u eksperimentima ovoga istraživanja rezultiralo je vrlo dobrim rezultatima, štoviše, izračun kosinusne sličnosti između vektorskih reprezentacija parova dokumenata modela obitelji BERT, čak i za velike korpuse poput Webis-11 s više od 15000 rečenica, koristeći prosječna računalima, prošlo je bez poteškoća i unutar nekoliko minuta (vidi tablicu 16), precizno: 86.4 sekunde. U isto vrijeme, neke standardne mjere za izračunavanje sličnosti nizova i riječi, poput *Levenshteina* ili *Greedy Word Tilinga*, vrlo su zahtjevne u pogledu potrebnih računalnih resursa i vremena potrebnog za njihove izračune.

5.5. Tehničke specifikacije

provedeni su na računalima s Debian OS verzijama od 9-11. Za dizajn eksperimenata korišten je programski jezik *Python* (verzije 2.7-3.9). *Python* se zajedno s *C++* smatra *de facto* standardom za NLP jer prevladava u broju i trendu implementacija. Na početku istraživanja mnoge potrebne biblioteke nisu još bile razvijene za tada novu verziju *Python 3*. Stoga je razvoj programa započeo s verzijom 2.7. S vremenom su potrebne biblioteke razvijene za *Python 3*, a prestale su se razvijati i postale zastarjele ili nedostupne za *Python 2*, pa je razvoj programa nastavljen u verziji 3. Korištene su mnoge biblioteke otvorenoga koda, među kojima su najvažnije *gensim*, *scipy*, *numpy*, *math*, *itertools*, *multiprocessing*, *matplotlib*,

nltk, *pickle, sklearn*, *nltk*, *pandas*, *tensorflow*, *transformers* i *cython*. Sve njih optimizirali su njihovi autori tako da u pozadini koriste *C++* kod. Unatoč tome te optimizacijama i višejezgrenoj obradi podataka, neke metode su se na većim korpusima izvršavale i po šest tjedana.

PyTorch i TensorFlow su programske biblioteke otvorenoga koda koje se koriste za duboko učenje. Omogućuju izgradnju, treniranje i implementaciju neuronskih mreža koristeći tenzore, automatsko diferenciranje i optimizatore, uz podršku GPU-a. PyTorch je razvio Facebookov AI Research lab (FAIR). Biblioteka je poznata po svojoj jednostavnosti korištenja, fleksibilnosti i dinamičkim računalnim grafovima. *PyTorch* je postao popularan alat među istraživačima i inženjerima za razvoj i implementaciju modela dubokog učenja (Paszke i sur., 2017; The Linux Foundation, 2023). Za razliku od statičkih grafova koje koristi *TensorFlow*, *PyTorch* koristi dinamičke računalne grafove koji se mogu mijenjati tijekom izvođenja. To omogućuje veću fleksibilnost prilikom eksperimentiranja i debagiranja modela. *PyTorch* je duboko integriran s Pythonom, što ga čini jednostavnim za korištenje za sve koji su upoznati s Pythonom. To omogućuje brzi razvoj i iteraciju modela. Pored toga, *PyTorch* pruža učinkovitu podršku za GPU-ove, omogućujući brzo treniranje modela i njihovu primjenu na nove podatke. *PyTorch* ima širok raspon alata i biblioteka koji pokrivaju različite aspekte dubokog učenja i aktivnu zajednicu razvijatelja i korisnika, što osigurava stalni razvoj i podršku. *PyTorch* je odličan izbor za istraživače i inženjere koji traže fleksibilan i moćan okvir za duboko učenje. Njegovi dinamički računalni grafovi, „pythonic" pristup i korištenje GPU-a za ubrzanje izračuna čine ga idealnim za brzo eksperimentiranje i razvoj najsuvremenijih modela. *TensorFlow* je razvio Google Brain tim (Abadi i sur., 2016) i temelji se na računalnim grafovima, gdje čvorovi predstavljaju matematičke operacije, a rubovi predstavljaju tenzore, višedimenzionalne nizove podataka. Ova struktura omogućuje učinkovito izvođenje složenih izračuna na različitim hardverskim platformama, uključujući CPU-ove, GPU-ove i TPU-ove. *TensorFlow* također podržava automatsko diferenciranje, ključnu tehniku za treniranje modela dubokog učenja izračunavanjem gradijenata (Vihar Kurama, 2024). *TensorFlow* podržava različite stilove programiranja, uključujući imperativno, deklarativno i funkcionalno programiranje, omogućujući korisnicima odabir pristupa koji najbolje odgovara njihovim potrebama (Abadi i sur., 2016). *TensorFlow* je vrlo skalabilan i može se skalirati od jednog uređaja do velikih klastera, što ga čini prikladnim za treniranje i primjenu velikih modela. *TensorFlow* nudi i opsežan ekosustav alata za

vizualizaciju, implementaciju modela za mobilne i ugrađene uređaje. *PyTorch* je pogodniji za istraživanje i brzo prototipiranje, a *TensorFlow* za velike projekte i proizvodnju.

Hugging Face[1] *Transformers* je također biblioteka otvorenoga koda koja omogućuje jednostavno korištenje predtreniranih modela i alata za izgradnju vlastitih modela (Hugging Face team, 2024; Vaswani i sur., 2017) jednostavnim API-jima i alatima za fino podešavanje. Transformer[2] je arhitektura neuronske mreže koja se temelji na mehanizmu pažnje, omogućujući modelima učinkovitu obradu slijednih podataka (Vaswani i sur., 2017). *Hugging Face Transformers* je biblioteka otvorenoga koda izgrađena povrh druge dvije biblioteke otvorenoga koda, *PyTorcha* i *TensorFlowa*, koja pruža implementaciju arhitekture transformera, zajedno s unaprijed treniranim modelima, alatima za fino podešavanje i jednostavnim API-jima. *Hugging Face Transformers* biblioteka pruža pristup širokom rasponu unaprijed treniranih modela za različite NLP zadatke, kao što su *BERT*, *GPT*, *T5* i *RoBERTa*.

Kao hardverska osnova izvršavanja programa Python, tj. eksperimenata, korištena su tri računala. Prvenstveno su korištena dva tipična prijenosna računala s dvojezgrenim i7 7. generacije i četverojezgrenim i5 10. generacije CPU-ima, s 20 GB i 12 GB RAM-a, bez dodatne podrške za paralelnu obradu (tj. s integriranom Intel HD grafikom, bez modernih GPU procesora s velikim brojem *Cuda* jezgri). Skroman hardver rezultirao je potrebom za maksimalnom optimizacijom programskoga koda. Jedan manji dio poslova obavljen je na radnoj stanici s 48 jezgri, s 2 TB RAM-a i tri *Nvidia Quadro RTX* 6000 24 GB GDDR6 grafičke kartice. Unatoč toj respektabilnoj snazi, za neke metode – parove (mjera, model) trebalo je puno vremena za njihovu obradu, pa je na kraju potvrđeno da su neke mjere udaljenosti (*Word Mover's Distance* i *Levenshtein*) i model koji se temelji na dubokom učenju *Embeddings from Language Models* (ELMo), prezahtjevni za današnju uobičajenu računalnu snagu, stoga su gotovo neupotrebljivi i treba ih odbaciti iz pragmatičnih razloga. Isto vrijedi i za par meke kosinusne sličnosti i *Glove Words* modela. Uz to rezultati tih metoda koje je radna stanica izračunala nakon nekoliko tjedana nisu vrhunski, pa njihovo odbacivanje i nije velik gubitak.

Eksperimenti **nad rečenicama** izvršeni su na prijenosnom računalu *Dell* G15 5511 s *Intel*(R) *Core*(TM) i7-11800H procesorom 11-te generacije s 8 jezgri (16 dretvi), 16GB

1 Hugging Face je platforma i zajednica za razvoj i dijeljenje modela strojnog učenja, dostupna na https://huggingface.co/.

2 Detaljnije u poglavlju 3.3.1. Arhitektura transformera.

RAM-a, *Nvidia GeForce RTX* 3060 grafičkom karticom (s 3840 Cuda jezgri) te operacijskim sustavom Debian GNU/Linux 12. Kodiranje je odrađeno u programskom jeziku Python verzije 3.

6. Rezultati

U ovome poglavlju opisani su postupci i rezultati vrednovanja provedenih eksperimenata. Najprije su u poglavlju 6.1. opisani rezultati eksperimenata za obradu teksta. Nakon toga su u poglavljima 6.2. i 6.3. opisani u postupci za vrednovanje detekcije sličnosti na razini dokumenata i vrednovanje detekcije parafraziranja na razini rečenice, a potom su u poglavljima 6.4., 6.5. i 6.6. predstavljeni rezultati tih postupaka. Na kraju, u poglavlju 6.7. predstavljena je konačna verzija metode DLPDM s parametrima koji su eksperimentalno utvrđeni kao najbolji. Kroz provedene eksperimente i evaluaciju koja je dio istraživačke metodologije ujedno je vrednovana predložena metoda.

6.1. Rezultati eksperimenata obrade teksta

U ovome poglavlju prikazani su rezultati eksperimenata u kojima su testirane različite tehnike za obradu i pripremu teksta. Uspoređivani su rezultati bez obrade i rezultati s različitim oblicima obrade, pri čemu je evaluacija provedena korištenjem F1-mjere kao balansirane metrike koja uzima u obzir točnost i odziv. U nastavku eksperimenata uvijek je ostavljana ona opcija koja je nakon evaluacije rezultata davala višu F1-mjeru. Posebno kod predtreniranih jezičnih modela temeljenih na dubokom učenju, rezultati su pokazali da minimalna obrada teksta često daje bolje rezultate. Predtrenirani jezični modeli trenirani su na velikim količinama teksta kojima nisu uklanjane zaustavne riječi (*stop-words*) ni vršene slične standardne obrade i transformacije ulaznog teksta već samo neznatne, nužne: tokenizacija, tokenizacija dijelova riječi, uklanjanje nerelevantnih znakova (posebnih simbola, nepotrebnih razmaka ili nekih stilskih elemenata), filtriranje neželjenih sadržaja, razdvajanje u rečenice i paragrafe (Brown i sur., 2020; Devlin i sur., 2018) pa su mogli sami naučiti prepoznati bitne informacije unatoč prisutnosti tih elemenata. Cilj je takve minimalističke obrade zadržavanje što je moguće više semantičkih informacija iz originalnog teksta. U nastavku su u tablicama 17-20 prikazani utjecaji pojedinih vrsta obrade teksta na rezultate.

Tablica 17. Utjecaj obrade teksta na uspješnost otkrivanja parafraziranja ukupno najuspješnijega modela mjereno F1-mjerom (korpus MSRP)

Metoda obrade teksta	Vrijednost	*θ*	*P*	*R*	F1	MCC
samo alfanumerički znakovi	True	0.67	0.721	0.966	0.826	0.370
	False	0.71	0.749	0.949	**0.837**	0.436
bigrami	True	0.69	0.736	0.960	0.833	0.411
	False	0.71	0.749	0.949	**0.837**	0.436
mala slova	**True**	0.71	0.749	0.949	**0.837**	0.436
	False	0.70	0.742	0.953	0.834	0.420
min. br. riječi	**1**	0.71	0.749	0.949	**0.837**	0.436
	3	0.62	0.703	0.956	0.810	0.285
	5	0.56	0.680	0.966	0.798	0.191
brojevi	**keep**	0.71	0.749	0.949	**0.837**	0.436
	delete	0.68	0.722	0.950	0.820	0.349
	tokens	0.69	0.706	0.964	0.815	0.311
riječi od 1 znaka	**True**	0.71	0.749	0.949	**0.837**	0.436
	False	0.71	0.747	0.941	0.833	0.420
korjenovanje	**None**	0.71	0.749	0.949	**0.837**	0.436
	Snowball	0.67	0.730	0.952	0.826	0.380
	Porter	0.69	0.738	0.942	0.828	0.394
	Porter2	0.67	0.730	0.952	0.826	0.380
stop riječi	**keep**	0.71	0.749	0.949	**0.837**	0.436
	delete	0.68	0.727	0.929	0.816	0.344
	replace	0.80	0.721	0.929	0.812	0.323
hiperonim/holonim	**None**	0.71	0.749	0.949	**0.837**	0.436
	hypernym	0.59	0.690	0.966	0.805	0.240
	holonym	0.64	0.721	0.959	0.823	0.359
lematizacija	**True**	0.71	0.749	0.949	**0.837**	0.436
	False	0.71	0.747	0.946	0.835	0.426
WordNet morfološke transformacije	True	0.74	0.761	0.909	0.829	0.422
	False	0.71	0.749	0.949	**0.837**	0.436

* Naziv modela: *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*

Tablice 17 i 18 prikazuju evaluirane rezultate koje postiže metoda najuspješnijeg modela *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*, tj. vektorske reprezentacije modela koje se uspoređuju putem kosinusne mjere sličnosti. Model je predtreniran, jedan od 149 preuzetih i korištenih tijekom istraživanja. U pravilu za te modele nije poznat (javno objavljen) način obrade teksta prije treniranja modela, znano je samo da je riječ o golemim skupovima probranih tekstnih dokumenata. Od mnoštva u istraživanju korištenih obrada samo je pretvaranje teksta u mala slova te lematizacija imala pozitivan učinak na rezultate. Pomalo su iznenađujući neki rezultati koji govore o tome da je pozitivan

učinak ostavljanja nealfanumeričkih znakova uključujući interpunkcijske, ostavljanja jednoznakovnih, pa čak i zaustavnih riječi (*stop-words*). No rezultati su konzistentni na dva korpusa (MSRP i P4PIN).

Rezultati evaluacije na temelju F1-mjere i MCC koeficijenta međusobno su prilično usklađeni, s time da i oni ukazuju da je P4PIN korpus znatno boljih službenih oznaka od MSRP. U ovoj analizi tablica 17 i 18 najvažnije je da se razne obrade teksta koje se smatraju standardnima prije treniranja jezičnih modela u ovim zadacima nisu pokazale primjerene, uz iznimku pretvorbe teksta u mala slova, tokenizacije i lematizacije.

Tablica 18. Utjecaj obrade teksta na uspješnost otkrivanja parafraziranja ukupno najuspješnijeg modela mjereno F1-mjerom (korpus P4PIN)

Metoda obrade teksta	Vrijednost	*θ*	*P*	*R*	F1	MCC
samo alfanumerički znakovi	True	0.76	0.962	0.939	0.950	0.935
	False	0.76	0.969	0.945	0.957	0.943
bigrami	True	0.76	0.957	0.951	0.954	0.939
	False	0.76	0.969	0.945	0.957	0.943
mala slova	**True**	0.76	0.969	0.945	0.957	0.943
	False	0.73	0.934	0.957	0.945	0.928
min. br. riječi	**1**	0.76	0.969	0.945	0.957	0.943
	3	0.74	0.925	0.834	0.877	0.843
	5	0.74	0.927	0.779	0.847	0.808
brojevi	**keep**	0.76	0.969	0.945	0.957	0.943
	delete	0.74	0.939	0.951	0.945	0.927
	tokens	0.76	0.957	0.945	0.951	0.935
riječi od 1 znaka	**True**	0.76	0.969	0.945	0.957	0.943
	False	0.73	0.934	0.957	0.945	0.928
korjenovanje	**None**	0.76	0.969	0.945	**0.957**	0.943
	Snowball	0.72	0.920	0.914	0.917	0.890
	Porter	0.71	0.888	0.920	0.904	0.872
	Porter2	0.72	0.920	0.914	0.917	0.890
stop riječi	**keep**	0.76	0.969	0.945	**0.957**	0.943
	delete	0.71	0.938	0.926	**0.932**	0.910
	replace	0.80	0.902	0.908	0.905	0.875
hiperonim/holonim	**None**	0.76	0.969	0.945	**0.957**	0.943
	hypernym	0.76	0.900	0.773	0.832	0.787
	holonym	0.76	0.927	0.859	0.892	0.860
lematizacija	**True**	0.76	0.969	0.945	**0.957**	0.943
	False	0.75	0.952	0.963	0.957	0.944
WordNet morfološke transformacije	True	0.74	0.963	0.945	0.954	0.939
	False	0.76	0.969	0.945	**0.957**	0.943

* Naziv modela: *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*

S obzirom na to da su u prethodne dvije tablice numerički prikazani utjecaji obrade teksta na rezultate predtreniranih jezičnih modela[1], postavlja se pitanje je li isti utjecaj i na jezične modele koji nisu predtrenirani, nego se za njihovo treniranje koristi korpus tekstova u kome se traže sličnosti. Stoga je isti postupak proveden i nad takvim jezičnim modelom. Odabran je onaj koji je nad MSRP i P4PIN korpusom prosječno dao najbolje rezultate, Doc2Vec Words[2] DBoW. S obzirom na to da su korišteni vektori riječi, vektori tekstova dobiveni su od vektora riječi kao normalizirana srednja vrijednost nenormaliziranih vektora riječi (vidi poglavlje *5.2.5. Konstrukcija vektorske reprezentacije teksta*). Iako Doc2Vec model ima mogućnost generiranja i vektora cijelih dokumenata/tekstova, rezultati istraživanja bili su bolji ako su vektori cijelih tekstova formirani na navedeni način.

U tablicama 19 i 20 predstavljeni su utjecaji različitih načina obrade teksta na rezultate mjerene F1-mjerom i MCC koeficijentom, na istovjetan način kao u prethodne dvije tablice, također na dva korpusa MSRP i P4PIN, ali korištenjem Doc2Vec Words DBoW modela koji je treniran na korpusima, a ne preuzet kao predtrenirani model.

Tablica 19. Utjecaj obrade teksta na uspješnost otkrivanja parafraziranja Doc2Vec Words DBoW modela treniranog na korpusu, mjereno F1-mjerom (korpus MSRP)

Metoda obrade teksta	Vrijednost	*θ*	*P*	*R*	F1	MCC
samo alfanumerički znakovi	True	0.49	0.685	0.960	0.799	0.208
	False	0.46	0.697	0.956	**0.806**	0.261
bigrami	True	0.00	0.657	0.985	0.788	0.027
	False	0.46	0.697	0.956	**0.806**	0.261
mala slova	**True**	0.40	0.674	0.994	0.803	0.211
	False	0.46	0.697	0.956	**0.806**	0.261
min. br. riječi	**1**	0.46	0.697	0.956	**0.806**	0.261
	3	0.21	0.663	0.986	0.793	0.100
	5	0.31	0.675	0.976	0.798	0.176
brojevi	**keep**	0.46	0.697	0.956	**0.806**	0.261
	delete	0.43	0.659	0.993	0.792	0.063
	tokens	0.35	0.660	0.991	0.792	0.077
riječi od 1 znaka	True	0.46	0.697	0.956	0.806	0.261
	False	0.43	0.699	0.957	**0.808**	0.273
korjenovanje	**None**	0.46	0.697	0.956	**0.806**	0.261
	Snowball	0.47	0.686	0.978	0.807	0.244
	Porter	0.52	0.699	0.960	0.809	0.274
	Porter2	0.44	0.684	0.968	0.802	0.216
stop riječi	**keep**	0.46	0.697	0.956	**0.806**	0.261

1 Svi se jezični modeli temelje na dubokom učenju, tako da se u tekstu taj izričaj skraćuje, tj. ne navodi.

2 *Words* je interna oznaka u istraživanju koja označava način dobivanja vektora cijelih dokumenata/tekstova putem vektora riječi, a ne putem vektorskih reprezentacija dokumenata koje stvara sam Doc2Vec model.

Metoda obrade teksta	Vrijednost	*θ*	*P*	*R*	F1	MCC
	delete	0.26	0.659	0.999	0.794	0.091
	replace	0.67	0.686	0.964	0.801	0.219
hiperonim/holonim	**None**	0.46	0.697	0.956	**0.806**	0.261
	hypernym	0.48	0.689	0.971	0.806	0.247
	holonym	0.49	0.703	0.944	0.806	0.270
lematizacija	**True**	0.46	0.697	0.956	**0.806**	0.261
	False	0.43	0.673	0.979	0.797	0.163
WordNet morfološke transformacije	True	0.46	0.687	0.959	0.801	0.219
	False	0.46	0.697	0.956	**0.806**	0.261

Iako su rezultati u tablicama 19 i 20 uglavnom u skladu s onima u tablicama 17 i 18, primjećuju se dvije razlike: Doc2Vec model treniran na ulaznom korpusu ostvarivao je bolje rezultate ako nije bilo pretvorbe teksta u mala slova te ako bi se iz teksta uklonile jednoznakovne riječi. Moguće objašnjenje za to bilo bi u golemoj razlici veličina korpusa tekstova kojim su trenirani modeli. Ostale preferencije u skladu su s onima predtreniranoga jezičnog modela, a razlike između *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs* i *Doc2Vec Words DBoW* modela prikazane su vrijednostima koje su istaknute crvenom bojom.

Tablica 20. Utjecaj obrade teksta na uspješnost otkrivanja parafraziranja modela Doc2Vec Words DBoW treniranoga na korpusu, mjereno F1-mjerom (korpus P4PIN)

Naziv	Vrijednost	*θ*	*P*	*R*	F1	MCC
samo alfanumerički znakovi	True	0.49	0.685	0.960	0.799	0.208
	False	0.46	0.697	0.956	**0.806**	0.261
bigrami	True	0.00	0.657	0.985	0.788	0.027
	False	0.46	0.697	0.956	**0.806**	0.261
mala slova	**True**	0.40	0.674	0.994	0.803	0.211
	False	0.46	0.697	0.956	**0.806**	0.261
min. br. riječi	**1**	0.46	0.697	0.956	**0.806**	0.261
	3	0.21	0.663	0.986	0.793	0.100
	5	0.31	0.675	0.976	0.798	0.176
brojevi	**keep**	0.46	0.697	0.956	**0.806**	0.261
	delete	0.43	0.659	0.993	0.792	0.063
	tokens	0.35	0.660	0.991	0.792	0.077
riječi od 1 znaka	True	0.46	0.697	0.956	0.806	0.261
	False	0.43	0.699	0.957	**0.808**	0.273
korjenovanje	**None**	0.46	0.697	0.956	**0.806**	0.261
	Snowball	0.47	0.686	0.978	0.807	0.244
	Porter	0.52	0.699	0.960	0.809	0.274
	Porter2	0.44	0.684	0.968	0.802	0.216

Naziv	Vrijednost	θ	P	R	F1	MCC
stop riječi	**keep**	0.46	0.697	0.956	**0.806**	0.261
	delete	0.26	0.659	0.999	0.794	0.091
	replace	0.67	0.686	0.964	0.801	0.219
hiperonim/holonim	**None**	0.46	0.697	0.956	**0.806**	0.261
	hypernym	0.48	0.689	0.971	0.806	0.247
	holonym	0.49	0.703	0.944	0.806	0.270
lematizacija	**True**	0.46	0.697	0.956	**0.806**	0.261
	False	0.43	0.673	0.979	0.797	0.163
WordNet morfološke transformacije	True	0.46	0.687	0.959	0.801	0.219
	False	0.46	0.697	0.956	**0.806**	0.261

Iako minimalna obrada teksta pokazuje dobre rezultate za jezične modele temeljene na dubokom učenju, u kontekstu manje složenih modela, određene tehnike obrade teksta mogu smanjiti varijabilnost podataka i poboljšati rezultate. Ipak, u provedenim eksperimentima za zadatak prepoznavanja parafraziranja pokazalo se da su takve tehnike obrade teksta ponekad kontraproduktivne. Na temelju rezultata provedenih eksperimenata zaključeno je da su tehnike obrade i pripreme teksta: tokenizacija (razdvajanje teksta na riječi ili rečenice), lematizacija (svođenje riječi na osnovni oblik) i pretvorba teksta u mala slova, odnosno definiran je niz O_{txt} na sljedeći način: O_{txt}=[*tokenizacija, lematizacija, pretvorba u mala slova*].

6.2. Evaluacija detekcije sličnosti na razini dokumenta

Evaluacija postupaka detekcije sličnosti na razini cjelovitih dokumenata provedena je kako bi se utvrdila učinkovitost primijenjenih metoda i jezičnih modela u prepoznavanju sličnosti između dokumenata unutar korpusa. Postupak uključuje binarizaciju rezultata sličnosti, odabir optimalnih mjera i modela te analizu performansi u više ciklusa usporedbe. Proces evaluacije obuhvaća sljedeće korake:

i) Evaluacija rezultata

Rezultati se evaluiraju za svaku graničnu vrijednost θ pomoću F1-mjere tako da je vrijednost θ ona kod koje je vrijednost funkcije F1 maksimalna, čime je određena i pripadna optimalna granična vrijednost za pojedinu metodu.

ii) Odabir najbolje metode prvog ciklusa (60 metoda)

Model M^* i granična vrijednost θ^* koji daju najbolji rezultat ulaze u sljedeću fazu.

Nakon što su sve metode iz prvog ciklusa evaluirane, identificirana je kosinusna sličnost s modelom BERT kao najbolja metoda.

iii) Dodatni ciklus (146 metoda)

Dodatni ciklus izračuna semantičke sličnosti te evaluacije rezultata imao je za cilj istražiti potencijalno bolje jezične modele srodne modelu BERT (modeli dubokog učenja nastali uglavnom prema arhitekturi transformera), koji je bio najbolje rangiran model iz prvog ciklusa. U ovome ciklusu na isti je način ispitano 146 jezičnih modela korištenjem kosinusne sličnosti kao mjere.

M_i={all-MiniLM-L12-v2, distilroberta-base-msmarco-v1, ..., aditeyabaral-roberta-base.}[1]

Za svaki model M_i iz skupa od 146 modela dubokog učenja i svaki par cjelovitih dokumenata (D, D_i) izračunata je semantička sličnost $sim_{\cos,Mi}$ (D, D_i), potom su rezultati evaluirani na istovjetan način kao u ciklusu sa 60 metoda. Nakon završetka oba ciklusa odabran je najbolji jezični model M^* iz dodatnog ciklusa evaluacije i njemu pripadni prag sličnosti θ^*. Oni se koriste za daljnju analizu u trećoj fazi gdje se uspoređuju rečenice (samo onih) dokumenata koji su se pokazali sličnim u smislu parafraziranja.

Dio postupka detekcije modela M^* napisan pseudokodom je sljedeći:

```
Za svaki jezični model Mi iz skupa od 146 modela:
    Za svaki dokument D u skupu dokumenata:
        Za svaki dokument Di iz korpusa C:
            Izračunaj kosinusnu sličnost između D i Di koristeći model Mi.
            Zabilježi rezultat sličnosti.
    Za svaki prag θ od 0 do 1 s korakom 0.01:
        Binariziraj rezultate sličnosti na temelju praga θ.
        Evaluiraj rezultate pomoću F-mjere i MCC.
        Zabilježi optimalnu graničnu vrijednost θ* za svaki model.
Odaberi najbolji model M* i njegov optimalni prag θ* koji daje najbolju F-mjeru.
Koristi najbolji model M* i pripadajući prag θ* za daljnju analizu na razini
rečenica.
```

6.2. Evaluacija postupka detekcije parafraziranja na razini rečenice

Evaluacija postupaka detekcije parafraziranja na razini rečenica provedena je radi

1 Vidi *Dodatak: Popis 146 modela drugog ciklusa eksperimenata (dokumenti)* za cjeloviti popis modela.

procjene učinkovitosti metode DLPDM i mjere DLCPM u prepoznavanju parafraziranih parova rečenica unutar dokumenata identificiranih kao sličnih. Postupak uključuje prikupljanje rezultata binarizacije i njihovu usporedbu sa službenim oznakama korpusa, uz primjenu evaluacijskih mjera za analizu performansi. Proces evaluacije obuhvaća sljedeće korake:

i) Prikupljanje rezultata za evaluaciju

Izračunata binarna vrijednost $b(s_j, s'_k)$ uspoređuje se s poznatom službenom oznakom $b_{official}(s_j, s'_k)$ koja označava je li par rečenica službeno označen kao parafraziran par (1) ili nije (0). Svi se rezultati prikupljaju u dvije liste: *calculated_values* ili *CV* – lista svih $b(s_j, s'_k)$ vrijednosti i *official_values* ili *OV* – lista svih $b_{official}(s_j, s'_k)$ vrijednosti.

ii) Evaluacija rezultata

Nakon što su sve vrijednosti prikupljene funkcije za evaluaciju računaju različite evaluacijske mjere, posebno F1-mjeru koja mjeri uspješnost klasifikacije i MCC koji mjeri kvalitetu binarne klasifikacije.

Dio postupka provjere napisan pseudokodom je sljedeći:

```
Za model M*:
    Za svaki par rečenica iz dvaju dokumenata:
        Izračunaj vrijednosti dlcpm mjere.
Za listu izračunatih vrijednosti dlcpm mjere:
    Za svaki prag θ od 0 do 1 s korakom 0.01:
        Binariziraj sličnosti na temelju praga θ.
        Izračunaj preciznost i odziv.
        Izračunaj F-mjeru:
            F-mjera = 2·(preciznost*odziv)/(preciznost+odziv)
        Izračunaj MCC (Matthewsov korelacijski koeficijent):
            MCC = (TP·TN-FP·FN)/sqrt((TP+FP)·(TP+FN)·(TN+FP)·(TN+FN))
        Zabilježi F-mjeru i MCC rezultate za svaki prag θ.
        Odaberi optimalnu graničnu vrijednost θ** koja daje
            najbolju F-mjeru.
```

6.4. Rezultati eksperimenata prvog ciklusa detekcije sličnosti na razini dokumenta

Rezultati prikazani u nastavku, u ovom i sljedećem poglavlju, rezultati su prve dvije faze eksperimenata sa 60 metoda i potom s njih još 146. Rezultati su dobiveni kao prosječne vrijednosti rezultata evaluacije za izračune izvršene za pet korpusa, dok su puni rezultati po korpusima prikazani u dodacima (*Dodatak: Cjeloviti rezultati prvog ciklusa eksperimenata (dokumenti)* i *Dodatak: Cjeloviti rezultati drugog ciklusa (dokumenti)*). U posljednjoj fazi, čiji su rezultati prikazani u poglavlju *6.6. Rezultati eksperimenata detekcije parafraziranja na razini rečenica*, u fazi u kojoj su uspoređivane rečenice, koristile su se samo tri metode na tri rečenična korpusa, stoga su za tu fazu prikazani cjeloviti rezultati. Rezultati su mjereni sljedećim evaluacijskim metrikama: preciznost (engl. *Precision*), odziv (engl. *Recall*), F1-mjera (engl. *F1-score*), Matthewsov korelacijski koeficijent (engl. *Matthews Correlation Coefficient*, MCC), točnost (engl. *Accuracy*) i specifičnost (engl. *Specificity*), pri čemu se F1-mjera koristila kao osnovna evaluacijska metrika i prema njoj su rangirani rezultati.

Ovo poglavlje prikazuje **ukupne prosječne rezultate eksperimenata prve faze** istraživanja, gdje je 60 metoda korišteno za izračun sličnosti dokumenata/tekstova iz pet korpusa. Većina predstavljenih metoda, njih 51, odnosi se na jezične modele uparene s mjerama sličnosti/udaljenosti, dok su ostale samostalne statističke mjere koje služe za usporedbu (engl. *baseline*), s obzirom na to da su to standardne metode koje su se koristile davno prije pojave jezičnih modela temeljenih na dubokom učenju.

Cilj prve faze istraživanja je utvrditi one mjere sličnosti i jezične modele s kojima će se, s obzirom na dobivene rezultate ići u daljnje istraživanje, s vizijom daljnjeg otkrivanja parafraziranja dijelova dokumenata/tekstova, tj. rečenica. Tablica 21 prikazuje prosječne rezultate eksperimenata prve faze tj. evaluaciju njihovih rezultata kroz pet korpusa, a detaljne tablice za svaki pojedini korpus nalaze se u poglavlju *Dodatak: Cjeloviti rezultati prvog ciklusa eksperimenata (dokumenti)*. Za svaku je metodu konačni rezultat dobiven kao aritmetička sredina rezultata (F-mjera, MCC) ostvarenih na korpusima CS, MSRP, P4PIN, VMEN i Webis. Za imenovanje metoda korištene su kratice samostalnih mjera ili kratice kombinirane od naziva mjere sličnosti (udaljenosti) i modela. Sve kratice objašnjene su u dodatku pod naslovom *Dodatak: Kratice*. Rezultati F1-mjere, kao primarnoga kriterija sortiranja rezultata, podebljani su, kao i redak s *cosbert* metodom koja je ukupno gledajući

najbolje rangirana u sumarnoj tablici 21. Najbolje rangirana metoda *cosbert* skraćena je oznaka za korištenje jednog od mnogih derivata jezičnog modela *BERT*, tj. njegovih vektorskih reprezentacija i kosinusne sličnosti kojom mjerimo sličnost tih vektora (kosinus kuta između vektora).

Tablica 21. Prosječna uspješnost 60 metoda na pet korpusa (sortirano prema F1 mjeri)

#	Metoda	θ	F	MCC
1	cosbert	0.59	0.867	0.676
2	edbert	0.36	0.865	0.675
3	mdbert	0.54	0.861	0.667
4	cosuse	0.59	0.854	0.658
5	eduse	0.37	0.852	0.657
6	mduse	0.55	0.852	0.655
7	le	0.22	0.848	0.642
8	tfidf	0.22	0.843	0.637
9	lsi	0.32	0.836	0.603
10	rp	0.23	0.833	0.609
11	jacc	0.16	0.833	0.609
12	wmdft	0.21	0.823	0.618
13	wmdd2vwdbow	0.18	0.815	0.597
14	wmdd2vwdm	0.18	0.815	0.598
15	wmdglw	0.18	0.812	0.595
16	coselmo	0.80	0.808	0.580
17	mdelmo	0.67	0.808	0.574
18	lev	0.24	0.807	0.536
19	edelmo	0.57	0.806	0.575
20	mdlaser	0.57	0.805	0.560
21	wmdw2vcbow	0.31	0.800	0.551
22	edlaser	0.49	0.785	0.559
23	coslaser	0.76	0.784	0.558
24	wmdw2vsg	0.37	0.782	0.566
25	scsd2vwdbow	0.54	0.761	0.548
26	scsd2vwdm	0.54	0.761	0.548
27	scsft	0.54	0.761	0.548
28	scsglw	0.54	0.761	0.548
29	scsw2vcbow	0.54	0.761	0.548
30	scsw2vsg	0.54	0.761	0.548
31	gwt	0.04	0.760	0.494
32	cosgld	0.42	0.749	0.527
33	edgld	0.24	0.725	0.498
34	edd2vwdbow	0.36	0.714	0.497
35	cosd2vwdbow	0.55	0.713	0.496
36	mdd2vwdbow	0.55	0.712	0.494
37	mdgld	0.47	0.711	0.475
38	cosd2vwdm	0.54	0.709	0.495
39	mdd2vwdm	0.49	0.702	0.487
40	cosglw	0.63	0.700	0.472
41	cosw2vcbow	0.55	0.696	0.478
42	edglw	0.38	0.694	0.471

#	Metoda	θ	F	MCC
43	mdglw	0.57	0.682	0.457
44	edw2vcbow	0.56	0.680	0.477
45	mdw2vcbow	0.69	0.655	0.453
46	cosft	0.61	0.622	0.415
47	edft	0.45	0.604	0.400
48	mdft	0.55	0.597	0.395
49	edd2vwdm	0.55	0.579	0.356
50	hdp	0.37	0.529	0.263
51	lda	0.44	0.525	0.224
52	cosw2vsg	0.82	0.519	0.352
53	mdw2vsg	0.76	0.513	0.332
54	edw2vsg	0.66	0.483	0.308
55	mdd2vddm	0.09	0.448	0.043
56	mdd2vddbow	0.09	0.447	0.041
57	cosd2vddm	0.03	0.363	-0.002
58	edd2vddm	0.02	0.363	-0.002
59	cosd2vddbow	0.04	0.350	0.002
60	edd2vddbow	0.02	0.347	-0.006

Najbolje rangirani prosječni rezultat (F1=0.867) eksperimenata prve faze nalazi se na vrhu tablice 21 i pripada metodi koja koristi *BERT* derivat *sentence-transformers-paraphrase-distilroberta-base-v1* uparen s kosinusnom mjerom sličnosti te vrijednošću praga binarizacije od 0.59. U pogledu odabira mjere sličnosti za buduće faze ovoga istraživanja to je predstavljalo jako dobar rezultat iz razloga što je računalni izračun kosinusne sličnosti iznimno brz i minimalno koristi računalne resurse. S obzirom na rezultate dobivene u prvoj fazi bilo je potrebno dodatno istražiti jezične modele u drugoj fazi jer metoda koja je polučila najbolje rezultate u prvoj fazi samo je predstavnik velike obitelji predtreniranih jezičnih modela temeljenih na dubokom učenju i u najvećoj mjeri arhitekturi transformera. Na drugom i trećem mjestu nalaze se srodne metode koje koriste isti jezični model, ali druge dvije mjere udaljenosti (Euklidsku i *Manhattan*).

Sljedeće tri metode koriste *USE* (F1=0.854), također jezični model koji se temelji na dubokom učenju, uparen s tri mjere sličnosti/udaljenosti, rangirane istim poretkom kao i za prvorangirani *sentence-transformers-paraphrase-distilroberta-base-v1* model. Iza tih šest metoda slijedi nekoliko statistički temeljenih mjera koje su dobile neočekivano dobre rezultate ($0.833 \leq F1 \leq 0.848$). Nasuprot tome neke su mjere imale niže rezultate od očekivanih, primjerice one temeljene na nizu znakova (engl. *string-based*), kao što su *Levenshtein* ili *Greedy Word Tiling* (GWT) i neki DL modeli poput Doc2Veca. Zanimljivo je da isti model DL može producirati značajno različite rezultate kada je uparen s različitim mjerama: *FastText*

s WMD (F1=0.823) u odnosu na *FastText* s *Manhattan* udaljenošću (F1=0.597); *Doc2Vec DBoW* s WMD (F1=0.815) naspram *Doc2Vec DBoW* s euklidskom udaljenošću (F1=0.347) i nekoliko drugih sličnih slučajeva. Ta teško objašnjiva činjenica bila je poticaj za dio istraživanja koji je trebao odgovoriti na pitanje koja je mjera sličnosti (udaljenosti) najbolja za otkrivanje parafraziranja.

Dakle, model s najboljom izvedbom korišten u prvoj fazi istraživanja jest model *sentence-transformers-paraphrase-distilroberta-base-v1*. Slobodno se može preuzeti iz repozitorija *Sentence Transformers*-a koji je *Python* okvir[1] za postizanje izvrsnih performansi na različitim zadacima (Reimers, 2021; Reimers i Gurevych, 2019). Modeli *Sentence Transformers* preslikavaju rečenice i odlomke u 768-dimenzionalni gusti vektorski prostor, a zatim se vektori iz njega koji predstavljaju određeni tekst mogu koristiti za zadatke poput klasteriranja ili semantičkog pretraživanja (Hugging Face, 2022). Platforma *Hugging Face*, poznata po svojoj biblioteci transformera, podržava izračun vektorskih reprezentacija rečenica/teksta za više od 100 jezika. Te se vektorske reprezentacije zatim mogu usporediti korištenjem neke mjere sličnosti ili udaljenosti kako bi se pronašle rečenice sličnoga značenja, tj. za primjenu u otkrivanju semantičke sličnosti, semantičkom pretraživanju ili u pronalaženju parafraziranja (Babić i sur., 2020; Reimers, 2021; Reimers i Gurevych, 2019). Modeli u okviru biblioteke *Sentence Transformers* trenirani su na milijunima parafraziranih rečenica kako bi prepoznavali semantičke sličnosti između različito formuliranih rečenic (Metatext, 2021).

Za daljnje eksperimente vrlo su važni i rezultati graničnih vrijednosti iz intervala [0, 1] dobivenih za svaku metodu, a koji predstavljaju optimalnu vrijednost binarizacije rezultata, kod njih je metoda najuspješnija u određivanju je li neki tekst parafraziran ili nije. Binarizacijom se stvarne vrijednosti kosinusne mjere sličnosti preslikavanju u binarni oblik [-1, 1] => {0, 1}, a radi usporedbe sa službenim oznakama korpusa parafraziranih tekstova, tj. evaluacije rezultata. Granična vrijednost utvrđena je tako što su rezultati za svaku metodu provlačeni kroz programsku petlju u intervalu [0,1] s korakom 0.01 te se za svaki od tih stotinu koraka računala F1-mjera. U konačnici je za svaku metodu odabrana ona granična vrijednost kod koje je F1-mjera postigla najveću vrijednost (za tu metodu).

1 Python okviri (engl. *framework*) su kolekcije modula i paketa koji programerima olakšavaju i automatiziraju implementaciju nekih (specijaliziranih) zadataka te im ubrzavaju proces programiranja.

6.5. Rezultati eksperimenata drugog ciklusa detekcije sličnosti na razini dokumenta

Ovo poglavlje prikazuje **ukupne prosječne rezultate druge faze** istraživanja, inspirirane najboljim rezultatom iz prve faze, u kojoj je identificiran model iz obitelji BERT (*sentence-transformers-paraphrase-distilroberta-base-v1*) kao najbolji. S obzirom na to da postoji mnogo javno dostupnih derivata BERT-a i drugih prethodno istreniranih jezičnih modela temeljenih na arhitekturi transformera, u drugoj fazi bilo je potrebno provjeriti njihov potencijal otkrivanja parafraziranja. Oni bi trebali imati različite potencijale za otkrivanje parafraziranja, neki možda i veće od modela korištenog u prvoj fazi. Bilo je stoga opravdano i potrebno istražiti sposobnosti svih dostupnih unaprijed treniranih jezičnih modela temeljenih na arhitekturi transformera, izrađenih s ciljem otkrivanja parafraziranja ili barem za otkrivanje sličnosti, te odrediti najbolji model za treću fazu istraživanja, koja ima za cilj pronaći parafrazirane rečenice prethodno otkrivenih sličnih dokumenata. Stoga su provedeni dodatni eksperimenti s unaprijed treniranim jezičnim modelima. Sažeti rezultati, ograničeni na najboljih 20 prema F1 vrijednostima, prikazani su u tablici 22, dok su puni rezultati za pojedine korpuse prikazani u pet tablica u poglavlju *Dodatak: Cjeloviti rezultati drugog ciklusa (dokumenti)*.

Tablica 22. Ukupna izvedba predtreniranih jezičnih modela (top 20) drugog ciklusa

#	Model	θ	F1	MCC
1	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.69	0.883	0.708
2	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.54	0.882	0.711
3	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.67	0.882	0.706
4	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.67	0.882	0.705
5	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.65	0.880	0.702
6	paraphrase-multilingual-mpnet-base-v2	0.65	0.880	0.702
7	paraphrase-mpnet-base-v2	0.63	0.880	0.702
8	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.66	0.878	0.698
9	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.61	0.877	0.688
10	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.61	0.877	0.688
11	paraphrase-multilingual-MiniLM-L12-v2	0.61	0.877	0.688
12	paraphrase-TinyBERT-L6-v2	0.58	0.877	0.688
13	paraphrase-distilroberta-base-v2	0.57	0.877	0.692
14	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.58	0.873	0.677
15	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.62	0.872	0.687
16	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.62	0.872	0.687
17	paraphrase-MiniLM-L12-v2	0.58	0.871	0.687
18	nli-mpnet-base-v2	0.73	0.871	0.684
19	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.61	0.871	0.680

20	multi-qa-mpnet-base-dot-v1	0.64	0.871	0.658

Prema prosječnim ukupnim rezultatima druge faze (rezultati pojedinih modela na pet korpusa prikazani su u poglavlju *Dodatak: Cjeloviti rezultati drugog ciklusa (dokumenti)*), najbolji model je *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs* s prosječnim F1=0.883. Model nije derivat BERT-a, poput BERT-a temelji se na arhitekturi transformera i treniran je posebno za zadatak generiranja parafraziranja, što uključuje generiranje rečenica ili fraza koje nose isto značenje kao i zadana ulazna rečenica ili fraza. Model je treniran na velikom višejezičnom korpusu i može generirati parafraze na više jezika. Model preslikava rečenice i odlomke u 768-dimenzionalni gusti vektorski prostor i može se koristiti za zadatke poput klasteriranja ili semantičkog pretraživanja (Farray, 2022). Taj se model temelji na višejezičnoj verziji modela MPNet (engl. *Megatron Pre-training*) koji je razvio tim *Hugging Face*. *Hugging Face* je popularna biblioteka otvorenoga koda za NLP koja pruža širok raspon predtreniranih modela za razne zadatke NLP-a, a jedan je od tih zadataka otkrivanje parafraziranja. Model MPNet je jezični model temeljen na arhitekturi transformera, sličan je BERT-u, ali koristi drukčiji pristup i arhitekturu prije treniranja. Autor modela vrlo je aktivan na *webu* zajednice *Hugging Face* s vlastitih prenesenih 36 modela (višejezičnih i španjolskih).

Na temelju opsežnih eksperimenata provedenih u dva ciklusa utvrđeni su konkretni parametri metode, M^* = *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*, *sim* = kosinusna_mjera_sličnosti i θ^*=0.69.

6.6. Rezultati eksperimenata detekcije parafraziranja na razini rečenica

U ovom su poglavlju predstavljeni **rezultati detekcije parafraziranja na razini rečenica**. U okviru provedenih eksperimenata uspoređeni su upareni tekstovi koji se sastoje od originalne i njoj parafrazirane rečenice, uz uvjet da za taj korpus postoje službene oznake. Za to je korištena nova kompozitna mjera parafraziranosti pomoću dubokog učenja, DLCPM. Ona je sastavljena od tri komponente. Prva komponenta koristi mjeru sličnosti koja je kombinacija kosinusne mjere sličnosti i vektora jezičnog modela. Korišten jezični model koji se temelji na dubokom učenju je model koji je imao najbolji ukupni rezultat iz prethodne faze usporedbe dokumenata *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*, uz njegovu pripadnu graničnu vrijednost. Izvorna optimalna granična vrijednost iz prethodne

faze istraživanja za navedeni jezični model iznosi 0.69, no zbog utjecaja druge dvije komponente i prirode kvadratne sredine koja se koristi za izračun DLCPM mjere, prag je eksperimentalnim rezultatima smanjen na 0.655. Druga komponenta nove DLCPM mjere je mjera sličnosti *Greedy Word Tiling*. Treća je komponenta nove DLCPM mjere sličnost dužine riječi dvaju tekstova.

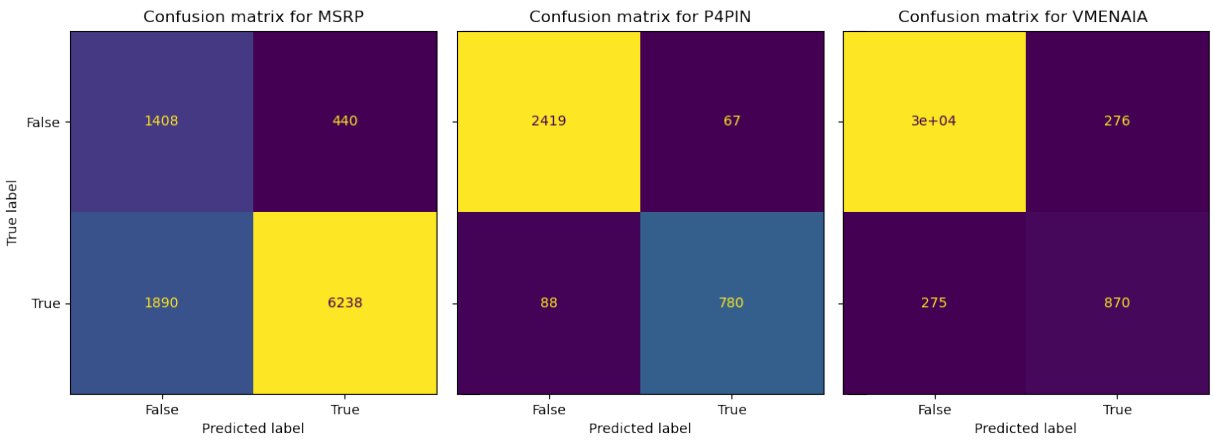
U prve dvije faze istraživanja i eksperimenata korišteno je 206 metoda (60 u prvoj i 146 u drugoj fazi), a najbolje od njih odabrane su za nastavak istraživanja na rečeničnoj razini. Ukupno je izvršeno 2053 eksperimenata koji su programski implementirani, provedeni, testirani i evaluirani. Kao što se naslućuje iz tablice 16, neki su eksperimenti trajali tjednima i mjesecima[1]. U trećoj fazi uspoređivani su manji tekstovi, tj. rečenice. Koristila se jedna mjera sličnosti i manji broj korpusa (tri umjesto pet). Stoga su za izvršenje zadataka u toj fazi bili potrebni znatno manja računalna snaga, radna memorija i kraća vremena izvršavanja eksperimenata nego kod usporedbe cjelovitih tekstova. To je omogućilo da se u trećoj fazi eksperimentira s različitim, do tada neistraženim, mogućnostima kompozitnog mjerenja i računanja ukupne sličnosti. Eksperimenti su provedeni nad tri rečenična korpusa parafraziranih parova tekstova: MSRP, P4PIN i VMENAIA, pri čemu je posljednji napravljen kao novi korpus za ovo istraživanje (vidi poglavlje *4.3.5. Korpusi VMEN i VMENAIA*), s obzirom na to da postoji nedostatak pouzdano kvalitetno označenih korpusa parafraziranih tekstova i da bi zaključivanje na dva postojeća možda bilo nedovoljno utemeljeno. Rezultati eksperimenata, tj. njihova evaluacija provedena je korištenjem standardnih evaluacijskih mjera kojima se evaluira sposobnost klasifikatora (koji u ovome slučaju klasificira rečenice na parafrazirane i one koje to nisu): F1-mjerom, MCC-om i mjerom sličnosti dužine riječi, a koje su već ranije objašnjene. Rezultati su prikazani tablicom 23.

Tablica 23. Rezultati evaluacije DLCPM mjere sličnosti na rečeničnim korpusima

Korpus	Odziv (*R*)	Preciznost (*P*)	F1-mjera (F1)	MCC
MSRP	0.7675	0.9341	0.8426	0.4372
P4PIN	0.8986	0.9209	0.9096	0.8787
VMENAIA	0.7598	0.7592	0.7595	0.7504

1 Najveći je dio istraživanja proveden na osrednjem sklopovlju, bez *Cuda* jezgri, tj. samo na CPU jezgrama (vidi hardversku osnovu eksperimenata pred kraj poglavlja *5.5. Tehničke specifikacije*). Prijenosno računalo s *Cuda* jezgrama korišteno pred kraj istraživanja upotrebljeno je za evidenciju vremena potrebnih za izvršenje eksperimenata i bez njega ne bi ni postojala *Tablica 16. Vrijeme izvršavanja metoda na podskupu train korpusa Webis-11,* jer je ono provodilo eksperimente 2-3 reda veličine brže od ranije korištenih računala.

Gledamo li rezultate iz tablice 23 kroz prizmu **preciznosti, odziva i F1-mjere**, rezultat za MSRP korpus ukazuje na poprilično dobru uravnoteženost između preciznosti (76.75%) i odziva (93.41%), a F1-mjera od 84.26% znači da DLCPM mjera na ovome korpusu vrlo dobro klasificira parafrazirane slučajeve, posebno pozitivne primjere. Dakle, ocjena DLCPM mjere parafraziranosti za MSRP korpus jest da je učinkovit, s jakom sposobnošću prepoznavanja stvarnih pozitivnih primjera i slabijom stvarnih negativnih, o čemu govori i matrica zbunjenosti na slici 26. Na P4PIN korpusu nova mjera postiže izvrsne rezultate s preciznošću, odzivom i F1-mjerom od oko i preko 90%, s vrlo dobrom ravnotežom između preciznosti i odziva, što čini DLCPM mjeru pouzdanom. DLCPM mjera na VMENAIA korpusu ima nešto slabije performanse u usporedbi s ostalim korpusima, ali je i dalje relativno solidna, s F1-mjerom od preko 75%, uz gotovo identične vrijednosti preciznosti, odziva i F1-mjere. Rezultati pokazuju da je DLCPM mjera u funkciji klasifikatora bolja u klasifikaciji parafraziranja P4PIN i MSRP korpusa, dok je malo slabija na VMENAIA korpusu. Objašnjenje koje se ovdje nameće jest da je to zbog obilježja VNEMAIA korpusa, tj. kvalitete (prikrivenosti) parafraziranog teksta u tom korpusu, povezanosti svake rečenice s više rečenica istog i sličnog dokumenta (parafraziranog ili originalnog pandana), kao i zbog nastanka korpusa tj. njegove pripadnosti istoj domeni (računalnim i informacijskim znanostima), pa je teže razlučivati bliske teme i njihove sastavnice – rečenice. Bez obzira na to rezultat F1-mjere od preko 75% pripada kategoriji vrlo dobrih klasifikatora, što je s obzirom na to da otkriva prikrivene plagijate nastale metodom parafraziranja, izvrstan rezultat.



Slika 26. Matrica zabune za mjeru DLCPM

Kao što je vidljivo iz matrice zabune za korpus MSRP na slici *26*, F1-mjera može biti visoka u situacijama gdje je model, tj. DLCPM mjera, dobra u prepoznavanju pozitivnih primjera, iako nije toliko uspješna u prepoznavanju negativnih. Za razliku od F1-mjere (Chicco i Jurman, 2020), MCC uzima u obzir sve četiri kategorije rezultata: stvarno pozitivne (*TP*), stvarno negativne (*TN*), lažno pozitivne (*FP*) i lažno negativne (*FN*). Stoga MCC može biti nizak ako model, u ovome slučaju mjera DLCPM nije uravnotežena u prepoznavanju i pozitivne i negativne klase.

U pogledu **MCC koeficijenta** koji ima vrijednosti iz intervala [-1,1], vrijednost za MSRP od 0.4372 ukazuje na pozitivnu korelaciju između stvarne i predviđene klasifikacije, iako nije iznimno jak. Taj rezultat sugerira da mjera DLCPM dobro razlikuje klase, no, kao što je spomenuto prije, mjera nije pohvalna u prepoznavanju stvarno negativnih veličina koje F1-mjera nije mogla registrirati. Za korpus P4PIN MCC rezultat od 0.8787 pokazuje da je mjera na tome korpusu vrlo učinkovita, s gotovo savršenom klasifikacijom. MCC rezultat od 0.7504 na VMENAIA korpusu znači da mjera ima vrlo dobre performanse.

Nova mjera DLCPM pokazala se učinkovitim klasifikatorom parafraziranja na korpusima s kvalitetnim podacima, poput P4PIN i VMENAIA. Korpus MSRP zahtijeva značajnu reviziju oznaka prije nego što se može koristiti za točnu evaluaciju modela, ili izbacivanje krivo označenih parova (primjeri krivih oznaka navedeni su u poglavlju *Dodatak: Primjeri krivih oznaka korpusa MSRP*), što u ovome istraživanju nije bilo uspješno na automatizirani način, a ručno mijenjanje/ispravljanje trajalo bi mjesecima[1]. To je i bio razlog potrebe za izradom novoga korpusa, kod kojega se mogu jamčiti dobre oznake. Performanse mjere DLCPM na korpusu MSRP ne odražavaju njegovu stvarnu sposobnost. Loši ulazni podaci ne mogu rezultirati dobrim izlaznim podacima; tako se i ovdje nova mjera DLCPM pokazala uspješnom kada se koristi s kvalitetnim podacima, a primjeri korpusa MSRP ukazuju na njegovu manjkavost.

Dakle, na temelju eksperimenata provedenih na razini rečenica, evaluirana je predložena mjera sličnosti DLCPM i utvrđen je konkretni parametar θ^{**}=0.655.

1 MSRP korpus je i koncepcijski barem djelomično promašen: sadrži jako puno burzovnih izvješća koji su redom vrlo slični i nemoguće je objektivno konstatirati za dva kratka burzovna izvješća o rastu ili padu cijena dionica iste tvrtke ili burzovnog indeksa jesu li parafraze ili ne.

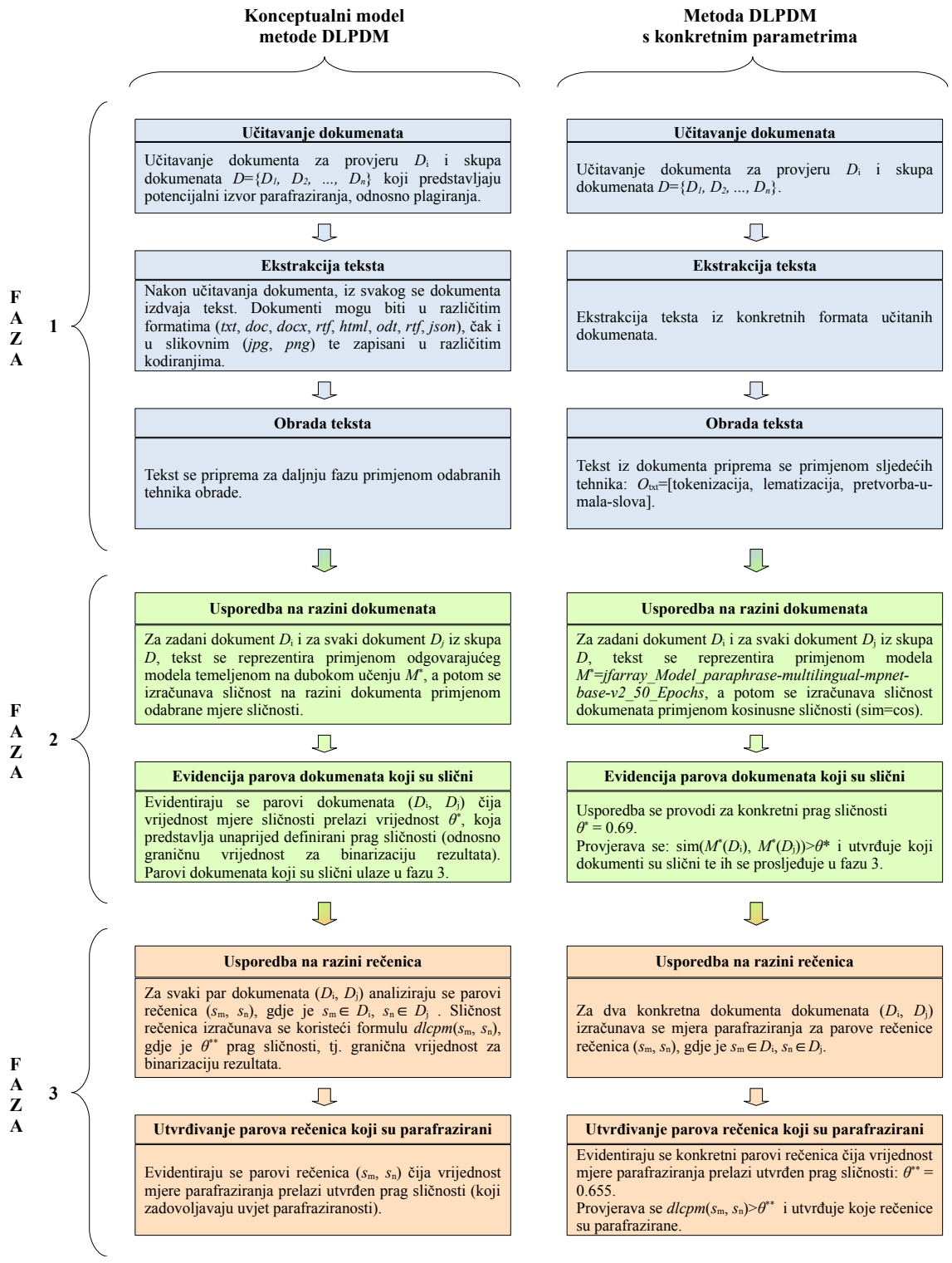
6.7. Metoda DLPDM

U ovom potpoglavlju ukratko su prezentirani oni parametri metode DLPDM koji su se kroz eksperimente pokazali najboljima. Metoda DLPDM je na konceptualnoj razini vrlo fleksibilna i sveobuhvatna u nekoliko aspekata (i) omogućava primjenu različitih tehnika pripreme teksta, (ii) omogućava korištenje različitih modela dubokog učenja i (iii) omogućava kombiniranje više sastavnica – rezultata metoda u traženju sinergijskog efekta za poboljšanje prepoznavanja parafraziranja. Predloženi konceptualni model metode DLPDM je modularan te se u budućnosti može mijenjati u odabranim elementima (promjena modela, promjena mjere sličnosti, promjena graničnih vrijednosti) ako se eksperimentalno utvrdi postojanje boljega, novijeg[1] elementa dok bi sama metoda na konceptualnoj razini ostala nepromijenjena.

Za potrebe ovoga istraživanja utvrđene su trenutno najbolje vrijednosti pojedinih elemenata na temelju dostupnih resursa (korpusi i modeli). Kao model koji najbolje detektira sličnost parafraziranih tekstova (M^*) identificiran je model *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*. Mjera sličnosti koja je dala najbolje rezultate u provedenim eksperimentima je kosinusna mjera sličnosti. Modelu svojstvene granične vrijednosti iznose θ^*=0.69 u fazi koja se odnosi na utvrđivanje sličnosti na razini dokumenata i θ^{**}=0.655 u fazi koja se odnosi na utvrđivanje parafraziranja na razini rečenica.

Na temelju dobivenih vrijednosti definirani su konkretni koraci metode kroz tri faze kako je prikazano na slici 27.

1 U trenutku pisanja već ima kandidata poput *GPT* i *DeepSeek* modela.



Slika 27. Opis metode DLPDM

Prva faza uključuje sljedeće korake: učitavanje korpusa, ekstrakcija teksta, obrada teksta. Druga faza podrazumijeva usporedbu i detekciju sličnosti na razini dokumenata

primjenom modela dubokog učenja *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs* i kosinusne mjere sličnosti. Ako je sličnost koja se računa primjenom mjere kosinusne sličnosti veća od zadanog praga θ^{*}=0.69, par dokumenata detektiran je kao sličan i ulazi u treću fazu. Treća faza podrazumijeva usporedbu i detekciju parafraziranja na razini rečenica za one dokumente koji su prešli zadani prag sličnosti primjenom mjere parafraziranja DLCPM. Preciznije, ako su rezultati mjere DLCPM pokazali da je sličnost između rečenica veća od zadanog praga θ^{**}=0.655 rečenice su proglašene parafraziranima.

Završna faza odnosi se na generiranje završnog izvještaja o parafraziranim dijelovima teksta.

7. Rasprava

U ovom je poglavlju prikazana analiza rezultata dobivenih u istraživanju, potom su raspravljena ograničenja istraživanja, nakon čega su predočeni znanstveni doprinosi te izneseni argumenti koji potvrđuju postavljene hipoteze.

Rezultati niza eksperimenata kroz sva istraživanja korišteni su za utvrđivanje elemenata metode DLPDM. Tako su detektirani jezični modeli s najboljim rezultatima, kako ukupnim za sve korpuse tako i one koji su najbolji za korpuse s manjim odnosno većim tekstovima. Stoga su ti najbolji modeli korišteni u drugoj fazi predložene metode DLPDM za detekciju parafraziranja na razini rečenica kao njezini sastavni elementi. Metoda je u tom smislu fleksibilna te je u svakom trenutku moguće jezične modele zamijeniti nekim budućima, boljima za pojedinu vrstu zadatka i ne nužno samo za detekciju parafraziranja.

U daljnjem tekstu prikazani su najbolji rezultati postignuti primjenom metode DLPDM za utvrđivanje parafraziranja dokumenata iz odabranih korpusa za koje postoje objavljeni rezultati drugih istraživanja. Rezultati su sagledani s aspekta detekcije sličnosti na razini dokumenta (druga faza metode) i detekcije parafraziranja na razini rečenice (treća faza metode) te su uspoređeni sa drugim istraživanjima.

Rezultati vrednovanja metode DLPDM na korpusu MSRP u zadatku detekcije sličnosti na razini dokumenta ima vrijednost F1-mjere od 84.4%, dok za detekciju parafraziranja na razini rečenice vrijednost F1-mjere iznosi 84.26%. Ti rezultati nadmašuju najbolje rezultate drugih istraživača (bez obzira na nadzirani ili nenadzirani pristup). MSRP (*Microsoft Research Parafrase Corpus*) široko je korišteni skup podataka (rečenični korpus) za

otkrivanje parafraziranja. Najbolji dosad objavljeni rezultati mjereni F1-mjerom iznose 80.2% (Rana i Mishra, 2019), 82% (Sánchez-Vega i sur., 2019; Saqaabi i sur., 2022), 83% (Álvarez-Carmona i sur., 2018), 83.71% (Saha i Kumar, 2021), 84.1% (Madnani i sur., 2012), što je i nešto više od teorijskog maksimuma od 83% koju je predstavila Mihalcea i sur. (2006) utvrdivši da je 83% gornja granica automatskog prepoznavanja parafraza na korpusu MSRP jer je to upravo onaj postotak u kojem su se sporazumjeli ljudi koji su označili parove kandidata parafraza (Mihalcea i sur., 2006). Vidljivo je da su neki istraživači prešli tu granicu ljudskog ocjenjivanja – zapravo je stvarna granica nepoznata i ovisi ponajviše o nepoznatoj kvaliteti korpusa i njegovih oznaka.

Na P4PIN korpusu u ovome istraživanju najbolji dobiven rezultat za F1-mjeru iznosi 97.8% (Vrbanec i Meštrović, 2023). Međutim treba napomenuti da je u drugoj fazi cilj bio otkriti najbolju metodu na pet korpusa za nastavak istraživanja (a ne na jednom od korpusa). Na temelju toga za nastavak istraživanja odabrana je metoda koja se pokazala prosječno najboljom za sve korpuse, što je primjerenije potrebama i situacijama stvarnoga svijeta. S tom prosječno najboljom metodom ostvaren rezultat F1-mjere iznosi 90.96%, a to je i dalje rezultat koji je u zadatku detekcije parafraziranja bolji od do sada najboljih. P4PIN je jezični korpus parafraziranih tekstova (rečenica) korišten u istraživanjima otkrivanja parafraziranja. Álvarez-Carmona i sur. (2018) korpus su podijelili na dijelove koji predstavljaju različite kategorije parafraziranja i zato nisu naveli sveobuhvatnu F1-mjeru, nego njezine vrijednosti za svaku od šest kategorija zasebno, pri čemu su u semantičkoj kategoriji parafraziranja, koja je jedina usporediva s ovdje predstavljenim istraživanjem, ostvarili rezultat F1-mjere od 78.9% (Álvarez-Carmona i sur., 2018). Drugi su istraživači na cjelokupnom korpusu dobili rezultat od 88.7% (Sánchez-Vega i sur., 2019).

Jedan od glavnih nedostataka ovoga istraživanja bio je manjak kvalitetno označenih korpusa parafraziranih tekstova. To je rezultirao je ograničenim istraživačkim mogućnostima u svim aspektima, pa tako i u pogledu usporedbe rezultata. Postoje rečenični korpusi načinjeni automatskim alatima, ali oznake sličnosti parova nisu binarni 0/1, već su to rezultati automatiziranih izračuna, decimalni brojevi koji ne pružaju mogućnost konstatacije stvarne istine (engl. *ground true*) kao temelj za usporedbu i evaluaciju rezultata jer za njih ne postoje poznate ili definirane granične vrijednosti za pretvorbu decimalnih vrijednosti u binarne vrijednosti.

Za potrebe ovoga istraživanja načinjen je originalni korpus pod nazivom VMENAIA,

tako da se za njega ne može provesti usporedba rezultata s prethodnim istraživanjima. Unatoč nemogućnosti usporedbe, valja ponoviti rezultat F1-mjere ovoga korpusa od 76%. VMENAIA je oblikovan na način da je višestruko koristan. S jedne strane, riječ je o 200 dokumenata, 100 sažetaka akademskih članaka na engleskom jeziku i isto toliko parafraziranih dokumenata, što znači da se može uspoređivati njihova sličnost na razini dokumenata. S druge strane, korpus je poravnat na način da i originalni i parafrazirani tekst imaju po jednak broj rečenica te je svaka rečenica pod istim rednim brojem uparena s njezinim pandanom (originalnom ili parafraziranom rečenicom); stoga se može raditi usporedba kao s rečeničnim korpusom, s time da je zadatak teži jer su svi dokumenti iz istih znanstvenih disciplina – računalnih i informacijskih znanosti.

7.1. Ostvareni znanstveni doprinosi

Tijekom istraživanja ostvarena su sva tri planirana odnosno očekivana znanstvena doprinosa, a ostvareno je i više dodatnih.

1. Oblikovan je korpus dokumenata pogodan za učenje i evaluaciju postupaka otkrivanja plagiranja pri parafraziranju. Načinjen je novi korpus parafraziranih tekstova koji se sastoji od 100 parova tekstova (izvorni, parafrazirani) sažetaka akademskih članaka, tako što su načinjeni parafrazirani tekstovi izvornih sažetaka, a potom su parafrazirani tekstovi dodatno parafrazirani uz pomoć dvaju generativnih jezičnih modela. Posljednja inačica korpusa dodatno je poravnata tako da parovi tekstova (izvorni i parafrazirani) imaju po jednak broj rečenica, odnosno da svaka rečenica izvornika ima svoje bijektivno preslikavanje u parafraziranu. Označavanje korpusa je implicitno, tj. rečenice iz istoga numeričkog para, s istim rednim brojem smatrane se parafraziranima, a sve ostale kombinacije nisu.

2. Razvijen je i implementiran novi postupak za otkrivanje plagiranja pri parafraziranju zasnovanog na modelu dubokog učenja. Istraživanjem je razvijena metoda DLPDM (detaljnije u *4. Istraživački postupak i razvoj metode DLPDM*) za otkrivanje plagijata temeljenih na parafraziranju. S tim je ciljem proveden opsežan skup eksperimenata za mjerenje semantičke sličnosti tekstova i uspoređene su performanse 60 metoda. Pri tome je korišteno pet korpusa parafraziranih tekstova, a rangiranje je određeno prema standardnim mjerama evaluacije. Eksperimentalni su rezultati u prvoj fazi pokazali da modeli dubokog

učenja nadmašuju statističke metode i da je jezični model iz obitelji BERT pod nazivom *distilroberta-base-paraphrase-v1* u paru s kosinusnom mjerom sličnosti najbolje rangirana metoda otkrivanja parafraziranja. Sposobnost otkrivanja parafraziranja na pet korpusa utvrđena je i za 149 unaprijed treniranih jezičnih modela, njih četiri u prvoj fazi (USE, ELMo, BERT i Laser) te 146 u drugoj (BERT[1] se preklapa u obje faze radi usporedbe). Utvrđeno je da model *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs* ima najbolji ukupni učinak prepoznavanja parafraziranja, mjereno F1-mjerom i koeficijentom MCC. Taj je model, pripadna kosinusna sličnost i pripadni prag preslikavanja iz decimalnog broja u binarnu oznaku (ocjena parafraziranosti 0/1), korišten za oblikovanje nove metode DLPDM za prepoznavanje parafraziranih rečenica u trećoj fazi, uz novomodeliranu DLCPM kompozitnu mjeru parafraziranosti pomoću jezičnih modela koji se temelje na dubokom učenju. Naime, nakon identifikacije parova dokumenata s dovoljnom sličnošću, metoda DLPDM predviđa fazu usporedbe rečenica – osnovnih jedinica teksta pomoću mjere DLCPM te ako njezina vrijednost za promatrani par rečenica premaši tijekom istraživanja utvrđen prag θ^*, ispitivana se rečenica u odnosu na uspoređivanu proglašava parafraziranom.

3. Definirani su postupak i mjera evaluacije parafraziranja tekstova. Tijekom istraživanja utvrđeno je da na konačni rezultat mjerenja parafraziranosti tekstova utječe više parametara i da rezultati mjere kosinusne sličnosti nad vektorskim reprezentacijama jezičnih modela temeljenih na dubokom učenju, iako već daju rezultate bolje od onih koje su dotad postigli drugi istraživači, mogu biti dodatno poboljšani. To je dodatno poboljšanje postignuto oblikovanjem nove DLCPM mjere parafraziranosti tekstova kao kompozitne mjere sastavljene od triju komponenata i ponderirane kvadratne sredine tih komponenata. Pri tome se koriste mjera sličnosti modela koja je pozitivno ponderirana, mjera sličnosti *Greedy Word Tiling* koja je negativno ponderirana i mjera sličnosti dužine riječi koja je neutralno ponderirana u ponderiranoj kvadratnoj sredini (vidi poglavlje *4.2.3. Istraživanje postupaka detekcije parafraziranja na razini rečenica*).

Uz to ostvareno je i nekoliko dodatnih znanstvenih doprinosa koji su opisani u nastavku.

1 Pod oznakom BERT u prvoj se fazi eksperimentiralo s nekoliko derivata BERT-a, a u konačnici je najbolji rezultat imao *distilroberta-base-paraphrase-v1*. U tim je eksperimentima postalo jasno da predtrenirani jezični modeli koji se temelje na dubokom učenju i uglavnom arhitekturi transformera imaju potencijal za bolje rezultate, što je istraženo i potvrđeno u drugoj fazi istraživanja na 146 u tom trenutku dostupnih modela, uključivši i *distilroberta-base-paraphrase-v1*.

Kroz niz provedenih eksperimenata utvrđene su tehnike obrade teksta koje pozitivno utječu na otkrivanje parafraziranja. Neke vrste obrade teksta utječu pozitivno na rezultate jezičnih modela dubokog učenja, a neke negativno, kako onih prethodno treniranih tako i onih koji se treniraju. U istraživanju je utvrđeno da kod zadatka detekcije parafraziranja na rezultate povoljno utječu pretvorba teksta u mala slova i lematizacija. Dakako da je i tokenizacija neophodna, dok sve ostale potencijalne obrade teksta imaju negativan utjecaj, što je opisano u poglavljima *5.1.1. Eksperimenti s tehnikama za obradu i pripremu teksta* i *6.1. Rezultati eksperimenata obrade teksta*).

Utvrđena je primjerenost različitih mjera sličnosti i udaljenosti za računanje sličnosti vektorskih reprezentacija tekstova. Analizirana je točnost pet mjera sličnosti i udaljenosti (kosinusna i meka kosinusna sličnost, te Euklidska, *Manhattan* i *Word Mover* udaljenost) za mjerenje sličnosti tekstova, a koje su primijenjene na izračun sličnosti ili udaljenosti vektorskih reprezentacija jezičnih modela temeljenih na dubokom učenju. Utvrđeno je da odabir mjere sličnosti, odnosno udaljenosti (u sprezi s vektorskim prostorima jezičnih modela) ne utječe dostatno na rezultate sličnosti da bi opravdalo korištenje značajnijih računalnih resursa od onih potrebnih za korištenje jednostavne kosinusne sličnosti. Također je utvrđeno da na rezultate mjera udaljenosti čije je rezultate potrebno transformirati u vrijednosti sličnosti uopće ne utječe način transformacije rezultata udaljenosti u rezultate sličnosti.

Procijenjena je složenost 60 metoda detekcije parafraziranja u *O* notaciji i predstavljena su vremena potrebna za provedbu jednoga reprezentativnog eksperimenta kao važnoga posrednog pokazatelja složenosti metoda.

Utvrđene su optimalne granične vrijednosti za binarizaciju metodama dobivenih decimalnih rezultata sličnosti tekstova u zadatku otkrivanja parafraziranja za mnoge metode utvrđivanja sličnosti tekstova. Binarizacijom se decimalne vrijednosti rezultata transformiraju u binarne vrijednosti 0 ili 1 koje govore o tome je li neki tekst parafraziran ili nije. Optimalne granične vrijednosti iznimno su bitne i svaka metoda ima svoju optimalnu graničnu vrijednost koja je istraživanjem egzaktno definirana.

Utvrđene su performanse metoda u zadatku detekcije parafraziranja temeljenih na korpusu. Rang-liste rezultata jasno su pokazale učinkovitost 60 metoda u zadatku detekcije parafraziranja te potom još 146 jezičnih modela na pet korpusa parafraziranih tekstova.

Definirana je formula kojom se aproksimira optimalni broj dimenzija vektorskih reprezentacija jezičnih modela dubokog učenja u zadatku detekcije parafraziranja. Za dva modela dubokog učenja (*Word2Vec* i *Doc2Vec*) koji su trenirani na dva korpusa provedena je detaljna analiza sposobnosti preslikavanja semantike teksta u njihove vektorske reprezentacije u ovisnosti o broju dimenzija vektorskih reprezentacija. Tako je za dimenzije vektorskih reprezentacija utvrđena funkcija koja aproksimira minimalno dostatan broj dimenzija (ili optimalan broj dimenzija), što može biti od velikog značaja i za treniranje velikih jezičnih modela u budućnosti, potencijalno znatno štedeći vrijeme i računalne resurse potrebne za njihovo treniranje.

7.2. Potvrda hipoteza

Na temelju predstavljenih podataka i prethodnih analiza, a glede postavljenih hipoteza, može se zaključiti da su obje hipoteze potvrđene kako slijedi u nastavku teksta.

Hipoteza H1: Primjenom modela dubokog učenja moguće je otkrivati plagiranje pri parafraziranju teksta.

Ta je hipoteza **potvrđena** kroz rezultate istraživanja. Korištenje jezičnih modela temeljenih na dubokom učenju, posebno onih iz obitelji BERT, njegovih derivata i jezičnih modela temeljenih na dubokom učenju i arhitekturi transformera, pokazalo je visoku uspješnost u otkrivanju parafraziranja, što je vidljivo iz rezultata provedenih eksperimenata. Modeli poput *sentence-transformers-paraphrase-distilroberta-base-v1* i *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs* ostvarili su vrlo visoke vrijednosti F1-mjere i MCC koeficijente, što ukazuje na njihovu sposobnost prepoznavanja semantičke sličnosti i parafraziranih tekstova. Rezultati treće faze, koji su se bavili detekcijom parafraziranja na razini rečenica, također su pokazali da modeli dubokog učenja uspješno klasificiraju parafrazirane rečenice, dosljedno ostvarujući visoke performanse na različitim korpusima, što ukazuje na njihovu robusnost.

Hipoteza H2: Odgovarajućom kombinacijom modela dubokog učenja i različitih tehnika pripreme teksta moguće je poboljšati otkrivanje parafraziranja.

Eksperimentalni rezultati **potvrđuju** drugu hipotezu (H2) da se kombinacijom modela dubokog učenja i odgovarajućih tehnika obrade teksta može dodatno poboljšati učinkovitost otkrivanja parafraziranja. Dok su neki klasični pristupi obrade, poput lematizacije, *stemminga*

i uklanjanja zaustavnih riječi (*stop-words*), često rezultirali slabijim performansama, određeni postupci poput pretvorbe teksta u mala slova, uklanjanje nepotrebnih razmaka i jednoznakovnih riječi, te lematizacija, imali su pozitivan učinak na učinkovitost modela u usporedbi s minimalnom obradom, posebno kod modernih, sofisticiranih, velikih, predtreniranih jezičnih modela temeljenih na dubokom učenju koji najuspješnije otkrivaju prikriveno plagiranje nastalo metodom parafraziranja (vidi poglavlja *5.1.1. Eksperimenti s tehnikama za obradu i pripremu teksta* i *6.1. Rezultati eksperimenata obrade teksta*, a posebno tablice 17-20). Ti postupci obrade omogućili su modelima preciznije prepoznavanje obrazaca parafraziranja i poboljšanje klasifikacije.

Posebno se ističe uvođenje DLCPM kompozitne mjere parafraziranosti, koja kombinira različite komponente mjera sličnosti tekstova. Ta mjera, u kombinaciji s modelima dubokog učenja, pokazala je značajna poboljšanja u prepoznavanju parafraziranih tekstova, što je vidljivo iz poboljšanja F1-mjere i koeficijenata MCC na korpusima P4PIN i VMENAIA. Istraživanje također potvrđuje da se, korištenjem specijaliziranih jezičnih modela i mjera sličnosti poput kosinusne sličnosti i GWT mjere, u kombinaciji s pravilnom pripremom teksta, mogu postići bolji rezultati u prepoznavanju parafraziranja (vidi poglavlje *6.6. Rezultati eksperimenata detekcije parafraziranja na razini rečenica*) . Stoga se može zaključiti da je druga hipoteza (H2) **potvrđena**.

Istraživanje jasno potvrđuje da su modeli dubokog učenja učinkoviti u otkrivanju parafraziranja (**potvrda H1**) te da kombinacija tih modela s odgovarajućim tehnikama pripreme teksta može poboljšati rezultate (**potvrda H2**)

8. Zaključak

Predstavljeno istraživanje rezultiralo je razvojem metode DLPDM za otkrivanje plagijata temeljenih na parafraziranju, s naglaskom na mjerenju semantičke sličnosti tekstova te korištenjem nove mjere DLCPM koja je u ovome radu predložena za mjerenje sličnosti parafraziranih tekstova. U istraživanju su provedeni opsežni eksperimenti na pet korpusa parafraziranih tekstova, gdje su ispitane performanse 209 metoda (60 u prvom i 146 u drugom ciklusu), uključujući modele dubokog učenja i statističke metode. Pokazano je da modeli dubokog učenja, posebno *distilroberta-base-paraphrase-v1* i *jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs*, pružaju najbolje rezultate, a utvrđene su i optimalne

granične vrijednosti potrebne za binarizaciju rezultata iz decimalnih u binarne brojeve koji označavaju klasifikaciju u kategoriju *parafrazirano*. Analizirano je kako različite mjere sličnosti i udaljenosti utječu na rezultate, pri čemu je utvrđeno da složenije mjere ne donose vidljivo poboljšanje u odnosu na jednostavnu mjeru sličnosti – kosinusnu sličnost, što predstavlja još jednu potvrdu ranije prakse i spoznaja. Također, stvoren je novi korpus parafraziranih tekstova VMENAI koji omogućuje preciznu evaluaciju metoda za prepoznavanje parafraziranja kako na razini cjelokupnih tekstova tako i na rečeničnoj razini. Korpus je stavljen u otvoreni pristup, čime su drugi istraživači dobili vrlo vrijedan resurs za svoja istraživanja. Proučena je složenost svih metoda te su predstavljena vremena koja su bila potrebna za izvođenje eksperimenata.

U prikazanom istraživanju prezentirana je i evaluirana nova kompozitna mjera parafraziranosti DLCPM temeljena na modelu dubokog učenja, sintetička mjera koja koristi ponderiranu kvadratnu sredinu i tri mjere sličnosti (kosinus kuta između vektora jezičnih modela, mjera sličnosti dobivena algoritmom *Greedy Word Tiling* te mjera sličnosti dužine riječi dvaju tekstova). Mjera izvrsno služi za otkrivanje parafraziranja[1], kao jednog od oblika prikrivenog plagiranja. Utvrđeno je i dokazano da je moguće efikasno i učinkovito otkrivati prikrivene plagijate koji su nastali parafraziranjem, s izvjesnošću da se opisana metoda može primijeniti i na druge oblike prikrivenog plagiranja, što bi se moglo pokazati u budućim istraživanjima. Akademska je zajednica time dobila snažan teorijski okvir i praktični temelj za izradu programskih alata za automatiziranu borbu protiv jedne od njezinih težih bolesti ili devijacije.

Kvaliteta ulaznih podataka ključni je čimbenik za točnost svih metoda, pa je potrebno više jezičnih resursa poput novoga, u radu predstavljenoga korpusa parafraziranih tekstova, na kojima bi se mogla provoditi buduća istraživanja otkrivanja prikrivenih plagijata, posebno onih nastalih parafraziranjem.

U budućim istraživanjima predložena metoda i mjera može se ispitati u višejezičnom kontekstu, tj. pored engleskog ispitati jezične modele temeljene na dubokom učenju s višejezičnošću ili specijalizirane za hrvatski jezik. Da bi hrvatsku inačicu metode bilo moguće evaluirati, prethodno je nužno poravnati HR korpus VMHR, koji je također načinjen tijekom

1 Mjera DLCPM u istraživanju se koristi na razini rečenica, no ona može poboljšati rezultate i na razini većih tekstnih cjelina, pa i cjelovitih tekstova. Razlog zašto se u istraživanju koristi samo na razini rečenica je dvojak: (a) mjera je otkrivena tek u posljednjoj fazi istraživanja – detekciji parafraziranja na razini rečenica i (b) zahtjevnost prema potrebnim računalnim resursima jedne njene komponente koja koristi *Greedy Word Tiling* algoritam.

istraživanja tako da se dobiju oznake parova rečenica *izvornik-parafraza* kako je to načinjeno i s novim, u istraživanju korištenim korpusom parafraziranih tekstova na engleskom jeziku, VMENAIA. U okviru budućih istraživanja planira se primjena novih velikih jezičnih modela koji se kontinuirano razvijaju i unaprjeđuju, poput *GPT*-a, *DeepSeek*-a ili *Grok*-a, s time da će se za potrebe istraživanja birati oni modeli koji su dostupni u otvorenom pristupu. Ti bi modeli, ako se pokažu boljima od onih koji su utvrđeni u ovome istraživanju, mogli također vrlo lako integrirati u metodu tako što bi zamijenili postojeće modele

Razvoj programskog dodatka za Moodle LMS jedna je od mogućih sljedećih etapa razvoja tj. budućih istraživanja, a njegova će se učinkovitost pokazati kroz evaluaciju studentskih radova. U konačnici, metoda i mjera mogu se koristiti i u drugim sustavima, primjerice kao programski dodatak za sustave kao što je *Open Journal Systems*, kojima se služi akademska zajednica kao podrškom za publiciranje akademskih radova u časopisima i na konferencijama, čime bi se urednicima časopisima i recenzentima dao uvid u eventualne parafrazirane tekstove, tj. one za koje postoji mogućnost da sadrže elemente prikrivenog plagiranja.

Literatura

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems* (No. arXiv:1603.04467). arXiv. https://doi.org/10.48550/arXiv.1603.04467

Abdelhamid, M., Azouaou, F., i Batata, S. (2022). *A Survey of Plagiarism Detection Systems: Case of Use with English, French and Arabic Languages* (No. arXiv:2201.03423). arXiv. http://arxiv.org/abs/2201.03423

Agarwal, B., Ramampiaro, H., Langseth, H., i Ruocco, M. (2017). A Deep Network Model for Paraphrase Detection in Short Text Messages. *Information Processing and Management*, *54*(6), 922–937. https://doi.org/10/gfdz6f

Ahmed, M., Samee, M. R., i Mercer, R. E. (2019). Improving Tree-LSTM with Tree Attention. *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/8665673/

Aiken, A. (2022). *MOSS: A System for Detecting Software Similarity*. https://theory.stanford.edu/~aiken/moss/

Aktion Plagiarius. (2018). *Innovation vs. Imitation*. https://www.plagiarius.com/index.php?ID=39

Ali, A., i Taqa, A. (2023). *Designing and Implementing Intelligent Textual Plagiarism Detection Models*.

Alpaydin, E. (2010). *Introduction to machine learning* (2nd ed). MIT Press.

Al-Shamery, E. S., i ALkhafaji, H. Q. G. (2017). Plagiarism and Source Deception Detection Based on Syntax Analysis. *Journal of University of Babylon*, *25*(2). https://www.iasj.net/iasj/download/1add287127b41fd9

Al-Shamery, E. S., i Gheni, H. Q. (2016). Plagiarism Detection using Semantic Analysis. *Indian Journal of Science and Technology*, *9*(1), 1–8. https://doi.org/10/ggv8c4

Alvarez, S. A. (2002). An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Boston College, Boston, Technical Report BCCS-02-01*, 1–22.

Álvarez-Carmona, M. A., Franco-Salvador, M., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., i Villaseñor-Pineda, L. (2018). Semantically-informed distance and similarity measures for paraphrase plagiarism identification. *Journal of Intelligent & Fuzzy Systems*, *34*(5), 2983–2990. https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169483

Alzahrani, S. M., Salim, N., i Abraham, A. (2012). Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(2), 133–149. https://doi.org/10/d4xx89

Amur, Z. H., Kwang Hooi, Y., Bhanbhro, H., Dahri, K., i Soomro, G. M. (2023). Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives. *Applied Sciences*, *13*(6), 3911. https://www.mdpi.com/2076-3417/13/6/3911

Anyscale. (2024). *Productionizing and scaling Python ML workloads simply*. Ray. https://www.ray.io/

Artetxe, M., i Schwenk, H. (2019). Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597–610. https://doi.org/10/gkz62j

Babić, K., Guerra, F., Martinčić-Ipšić, S., i Meštrović, A. (2020). A comparison of approaches for measuring the semantic similarity of short texts based on word embeddings. *Journal of Information and Organizational Sciences*, *44*(2), 231–246. https://doi.org/10.31341/jios.44.2.2

Babić, K., Martinšić-Ipšić, S., Meštrović, A., i Guerra, F. (2019). Short texts semantic similarity based on word embeddings. *Central European Conference on Information and Intelligent Systems*, 27–33.

Bailey, J. (2011, rujan 6). PlagScan Review: Solid Plagiarism Detection. *Plagiarism Today*. https://www.plagiarismtoday.com/2011/09/06/plagscan-review-solid-plagiarism-detection/

Bali, A., Bhagwat, A., Bhise, A., i Joshi, S. (2024). Semantic Similarity Detection and Analysis For Text Documents. *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, 1–9. https://doi.org/10.1109/ic-ETITE58242.2024.10493834

Banea, C., Chen, D., Mihalcea, R., Cardie, C., i Wiebe, J. (2014). Simcompass: Using deep learning word embeddings to assess cross-level similarity. *SemEval 2014*, 560–565.

http://www.aclweb.org/anthology/S14-2098

Bassil, Y., i Semaan, P. (2012). *Semantic-Sensitive Web Information Retrieval Model for HTML Documents* (No. arXiv:1204.0186). arXiv. https://doi.org/10.48550/arXiv.1204.0186

Bašić, B. D., i Šnajder, J. (2011). *Vrednovanje klasifikatora—Presentation lecture notes*.

Beames, S. (2012). *White Paper—The Plagiarism Spectrum: Instructor Insights into the Ten Types of Plagiarism*. 18.

Belinkov, Y., i Glass, J. (2019). Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, *7*, 49–72. https://doi.org/10.1162/tacl_a_00254

Bengio, Y., Ducharme, R., Vincent, P., i Jauvin, C. (2003). *A Neural Probabilistic Language Model*.

Birhane, A., Kasirzadeh, A., Leslie, D., i Wachter, S. (2023). Science in the age of large language models. *Nature Reviews Physics*, *5*(5), 277–280. https://doi.org/10.1038/s42254-023-00581-4

Blei, D. M., Ng, A. Y., i Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022. https://www.jmlr.org/papers/v3/blei03a

Bojanowski, P., Grave, E., Joulin, A., i Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *Computing Research Repository*. https://doi.org/1511.09249v1

Bouville, M. (2008). Plagiarism: Words and Ideas. *Science and Engineering Ethics*, *14*(3), 311–322. https://doi.org/10/dts9bv

Brennan, M. R., i Greenstadt, R. (2009). *Practical Attacks Against Authorship Recognition Techniques*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in neural information processing systems*. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G. C., Steinhardt, J., Flynn, C., hÉigeartaigh, S. Ó., Beard, S., Belfield, H., Farquhar, S., … Amodei, D. (2018). *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation* (No. arXiv:1802.07228). arXiv. https://doi.org/10.48550/arXiv.1802.07228

Budanitsky, A., i Hirst, G. (2001). Semantic Distance in Wordnet: An Experimental, Application-Oriented Evaluation of Five Measures. *Workshop on WordNet and Other Lexical Resources*.

Burrows, S., Potthast, M., Stein, B., i Eiselt, A. (2013, lipanj 1). *Webis Crowd Paraphrase Corpus 2011 (Webis-CPC-11)*. https://doi.org/10.5281/ZENODO.3251771

Callison-Burch, C. (2008). Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. *Proceedings of the Conference on Empirical Methods in Natural Language*

Processing, *October*, 196–205.

Cambridge University Press. (2018). *Meaning of "plagiarize" in the English Dictionary*. http://dictionary.cambridge.org/dictionary/english/plagiarize?q=plagiarism

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., i Kurzweil, R. (2018). Universal Sentence Encoder for English. U E. Blanco i W. Lu (Ur.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (str. 169–174). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-2029

Cer, D., Yang, Y., Kong, S. yi, Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Yun-Hsuan Sung, Strope, B., i Kurzweil, R. (2018). Universal Sentence Encoder. *arXiv:1803.11175 [cs.CL]*. http://arxiv.org/abs/1803.11175

Chandrasekaran, D., i Mago, V. (2022). Evolution of Semantic Similarity—A Survey. *ACM Computing Surveys*, *54*(2), 1–37. https://doi.org/10.1145/3440755

Chaudhuri, J. (2008). Deterring digital plagiarism, how effective is the digital detection process? *Webology*, *5*(1). http://www.webology.org/2008/v5n1/a50.html

Chicco, D., i Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Chong, M., Specia, L., i Mitkov, R. (2010). Using natural language processing for automatic detection of plagiarism. *Proceedings of the 4th International Plagiarism Conference (IPC 2010), Newcastle, UK*.

Chong, M. Y. M. (2013). *A study on plagiarism detection and plagiarism direction identification using natural language processing techniques* [University of Wolverhampton]. http://wlv.openrepository.com/wlv/handle/2436/298219

Clough, P. (2000). Plagiarism in natural and programming languages: An overview of current tools and technologies. *Research Memoranda: CS-00-05, Department of Computer Science, University of Sheffield, UK*, 1–31.

Clough, P., i Stevenson, M. (2009). Creating a Corpus of Plagiarised Academic Texts. *Proceedings of the Corpus Linguistics Conference*, *January 2009*.

Clough, P., i Stevenson, M. (2011). Developing a corpus of plagiarised short answers. *Language Resources and Evaluation*, *45*(1), 5–24. https://doi.org/10/bsnn7w

Corley, C., Csomai, A., i Mihalcea, R. (2007). A Knowledge-based Approach to Text-to-Text Similarity. *Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005*, *292*, 210–219.

Corley, C., i Mihalcea, R. (2005). Measuring the semantic similarity of texts. *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 13–18. https://doi.org/10/cpd93r

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Culwin, F., i Lancaster, T. (2001a). Plagiarism issues for higher education. *VINE*, *31*(2), 36–41. https://doi.org/10/fnn9zf

Culwin, F., i Lancaster, T. (2001b). Plagiarism, prevention, deterrence & detection. *Available for ILT members from*.

Czerski, D., Lozinski, P., Cacko, A., Szmit, R., i Tartanus, B. (2015). Fast plagiarism detection in large scale data. *International Conference on Tools, Applications and Implementations of Methods for Determining Similarities Between Documents (TAIM4DSBD)*, 1–4.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., i Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Deisenroth, M. P., Faisal, A. A., i Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press. https://books.google.com/books?hl=hr&lr=&id=pFjPDwAAQBAJ&oi=fnd&pg=PR9&dq=%22Mathematics+for+Machine+Learning%22+by+Marc+Peter+Deisenroth,+A.+Aldo+Faisal,+and+Cheng+Soon+Ong&ots=VMhq3FM5Ca&sig=I310Z0BReP1Aw5UtPopcTAsvc00

Devlin, J., Chang, M.-W., Lee, K., i Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. http://arxiv.org/abs/1810.04805

Dolan, W. B., i Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

El Desouki, M. I., Gomaa, W. H., i Abdalhakim, H. (2019). A Hybrid Model for Paraphrase Detection Combines pros of Text Similarity with Deep Learning. *International Journal of Computer Applications*, *178*(20), 18–23. https://doi.org/10/ghjqjk

El Mostafa, H., i Benabbou, F. (2020). A deep learning based technique for plagiarism detection: A comparative study. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *9*(1), 81. https://doi.org/10/ghjqhn

Eliseev, D. Y., i Sknarina, N. A. (2012). Postroenie modeli nechetkogo vyvoda pri organizacii avtomatizirovannoj besprovodnoj sistemy monitoringa ob"ektov zheleznodorozhnogo transporta. *Vestnik Rostovskogo gosudarstvennogo universiteta putej soobshcheniya*, *3*, 49–53. https://elibrary.ru/item.asp?id=17928781

Encyclopaedia Britannica. (2024, siječanj 29). *Plagiarism*. https://www.britannica.com/topic/plagiarism

Eshet, Y. (2024). The plagiarism pandemic: Inspection of academic dishonesty during the COVID-19 outbreak using originality software. *Education and Information Technologies*, *29*(3), 3279–3299. https://doi.org/10.1007/s10639-023-11967-3

Farray, J. (2022, travanj 25). *Jfarray (Jacobo Farray)*. https://huggingface.co/jfarray

Fedus, W., Zoph, B., i Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, *23*(120), 1–39. https://www.jmlr.org/papers/v23/21-0998.html

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., i Wang, W. (2020). *Language-agnostic BERT*

Sentence Embedding. http://arxiv.org/abs/2007.01852

Fernando, S., i Stevenson, M. (2008). A semantic similarity approach to paraphrase detection. *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, 45–52. https://doi.org/10.1.1.144.4680

Gao, X.-Z., Kumar, R., Srivastava, S., i Soni, B. P. (Ur.). (2021). *Applications of Artificial Intelligence in Engineering: Proceedings of First Global Conference on Artificial Intelligence and Applications (GCAIA 2020)*. Springer Singapore. https://doi.org/10.1007/978-981-33-4604-8

Gharavi, E., Bijari, K., Zahirnia, K., i Veisi, H. (2016). A Deep Learning Approach to Persian Plagiarism Detection. *FIRE 2016 - Forum for Information Retrieval Evaluation*, *34*, 154–159. https://www.academia.edu/download/78065131/T4-4.pdf

Gibbs, A. L., i Su, F. E. (2002). *On choosing and bounding probability metrics* (No. arXiv:math/0209021). arXiv. https://doi.org/10.48550/arXiv.math/0209021

Gipp, B. (2014). *Citation-based Plagiarism Detection—Detecting Disguised and Cross-language Plagiarism using Citation Pattern Analysis* (Sv. 9783658063). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-06394-8

Gipp, B., Meuschke, N., i Beel, J. (2011). Comparative evaluation of text-and citation-based plagiarism detection approaches using guttenplag. *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, 255–258. https://doi.org/10/d6q5xt

Gomaa, W. H., i Fahmy, A. A. (2017). SimAll: A flexible tool for text similarity. *The Seventeenth Conference On Language Engineering ESOLEC*, *17*(1), 122–127.

Goodfellow, I., Bengio, Y., i Courville, A. (2016). *Deep learning* (Sv. 22). MIT Press. https://books.google.com/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=Ian+Goodfellow+and+Yoshua+Bengio+and+Aaron+Courville&ots=MLU59rozNU&sig=YHYf6iAwhmFAQkBFuLnHAgrkMts

Google. (2023). *Gemini*. Gemini. https://gemini.google.com

Goutte, C., i Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. *European Conference on Information Retrieval*, 345–359. http://link.springer.com/10.1007%2F978-3-540-31865-1_25

Green, S. P. (2002). Plagiarism, Norms, and the Limits of Theft Law: Some Observations on the Use of Criminal Sanctions in Enforcing Intellectual Property Rights. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.315562

Han, M., Zhang, X., Yuan, X., Jiang, J., Yun, W., i Gao, C. (2021). A survey on the techniques, applications, and performance of short text semantic similarity. *Concurrency and Computation: Practice and Experience*, *33*(5), e5971. https://doi.org/10.1002/cpe.5971

Harispe, S., Ranwez, S., Janaqi, S., i Montmain, J. (2013). Semantic Measures for the Comparison of Units of Language, Concepts or Entities from Text and Knowledge Base Analysis. *arXiv preprint arXiv: ...*, *May*, 1–102. https://doi.org/10/gj26v8

Harispe, S., Ranwez, S., Janaqi, S., i Montmain, J. (2017). Semantic Similarity from Natural

Language and Ontology Analysis. *Synthesis Lectures on Human Language Technologies*, *8*(1), 1–254. https://doi.org/10/gc3jtd

Harispe, S., Sánchez, D., Ranwez, S., Janaqi, S., i Montmain, J. (2014). A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. *Journal of Biomedical Informatics*, *48*, 38–53. https://doi.org/10/f52557

Hoad, T., i Zobel, J. (2003). Methods for Identifying Versioned and Plagiarized Documents. *JASIST*, *54*, 203–215. https://doi.org/10.1002/asi.10170

Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*, 107–116. https://doi.org/10.1142/S0218488598000094

HRČAK. (2021). *Hrčak—Portal hrvatskih znanstvenih i stručnih časopisa*. https://hrcak.srce.hr/

Hrvatski Sabor. (2021, listopad 14). *Zakon o autorskom pravu i srodnim pravima, NN 111/2021*. Hrvatski Sabor. https://narodne-novine.nn.hr/clanci/sluzbeni/2021_10_111_1941.html

Hsiao, D. K., Neuhold, E. J., i Sacks-Davis, R. (2014). So far (schematically) yet so near (semantically). *Interoperable Database Systems (DS-5): Proceedings of the IFIP WG2. 6 Database Semantics Conference on Interoperable Database Systems (DS-5) Lorne, Victoria, Australia, 16-20 November, 1992*, *25*, 283.

Huang, A. (2008). Similarity measures for text document clustering. *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, *4*, 9–56.

Hugging Face. (2022). *Models—Hugging Face*. https://huggingface.co/models

Hugging Face team. (2024). *Transformers*. https://huggingface.co/docs/transformers/index

Jaccard, P. (1912). The Distribution of the Flora in the Alpine Zone. *New Phytologist*, *11*(2), 37–50. https://doi.org/10/fvhsjd

Jagtap, R. (2020, kolovoz 27). *LaBSE: Language-Agnostic BERT Sentence Embedding by Google AI*. Medium. https://towardsdatascience.com/labse-language-agnostic-bert-sentence-embedding-by-google-ai-531f677d775f

Jakupović, A., Pavlić, M., Meštrović, A., i Jovanović, V. (2013). Comparison of the Nodes of Knowledge method with other graphical methods for knowledge representation. *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on*, 1004–1008.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., i Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv:1909.10351 [cs]*. http://arxiv.org/abs/1909.10351

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., i Levy, O. (2019). SpanBERT: Improving Pre-training by Representing and Predicting Spans. *arxiv.org*. https://github.com/facebookresearch/

Joulin, A., Grave, E., Bojanowski, P., i Mikolov, T. (2016). *Bag of Tricks for Efficient Text Classification*. https://doi.org/1511.09249v1

Joy, M., Cosma, G., Sinclair, J., i Yau, J. (2009). A taxonomy of plagiarism in computer

science. *Proceedings of EDULEARN09 Conference: 6th - 8th July 2009*.

Juričić, V. (2012). *Detekcija plagijata u višejezičnom okruženju: Doktorska disertacija*.

Kakkonen, T., i Mozgovoy, M. (2010). Hermetic and Web Plagiarism Detection Systems for Student Essays—An Evaluation of the State-of-the-Art. *Journal of Educational Computing Research*, *42*(2), 135–159. https://doi.org/10/dr3789

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Naacl-2015*, 901–911. https://doi.org/10/ghjqhw

Knuth, D. E. (1989). The art of computer programming. *Seminumercial Algorithms*, *2*, 268–278. https://www.academia.edu/download/60755625/Art_of_computing_Algorithms20191001-98023-uzae1x.pdf

Koch, M. R. R., Pavlić, M., i Jakupović, A. (2014). Application of the NOK method in sentence modelling. *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, 1176–1181. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6859746

Kong, L., Lu, Z., Qi, H., i Han, Z. (2014). Detecting High Obfuscation Plagiarism: Exploring Multi-Features Fusion via Machine Learning. *International Journal of u-and e-Services, Science and Technology*, *7*(4), 385–396. https://doi.org/10/ggwpgg

Krause, E. F. (1986). *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation.

Krizhevsky, A., Sutskever, I., i Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kumar, R., i Tripathi, R. C. (2013). An Analysis of Automated Detection Techniques for Textual Similarity in Research Documents. *International Journal of Advanced Science and Technology*, *56*, 99–110.

Kusner, M. J., Sun, Y., Kolkin, N. I., i Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. *Proceedings of the International Conference on Machine Learning 2015*, 957–966. https://www.semanticscholar.org/paper/From-Word-Embeddings-To-Document-Distances-Kusner-Sun/66021a920001bc3e6258bffe7076d647614147b7

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., i Research, G. (2020). Albert: A lite bert for self-supervised learning of language representations. *ICLR 2020*. https://github.com/google-research/ALBERT.

Lancaster, T. (2003). *Effective and efficient plagiarism detection* [PhD Thesis, London South Bank University London, UK]. https://www.researchgate.net/profile/Thomas-Lancaster-2/publication/228729388_Effective_and_Efficient_Plagiarism_Detection/links/0fcfd50f68dcf52345000000/Effective-and-Efficient-Plagiarism-Detection.pdf

Lancaster, T., i Culwin, F. (2005). Classifications of plagiarism detection engines. *Innovation in Teaching and Learning in Information and Computer Sciences*, *4*(2). https://doi.org/10/gj26t7

Landauer, T. K., Foltz, P. W., i Laham, D. (1998). An introduction to latent semantic analysis.

Discourse Processes, *25*(2–3), 259–284. https://doi.org/10/fsc78g

Le, Q. V., i Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Machine Learning*, *14*, 1188–1196.

Leacock, C., Chodorow, M., i Miller, G. A. (1998). *Combining local context and WordNet sense similarity for word sense identification* (C. Fellbaum, Ur.; str. 265–283). The MIT Press Cambridge.

Lee, C. (2020, kolovoz 10). *What Are the New and Emerging Plagiarism Trends?* https://www.turnitin.com/blog/what-are-the-new-and-emerging-plagiarism-trends

Lee, M. D., Pincombe, B., i Welsh, M. (2005). An Empirical Evaluation of Models of Text Document Similarity. *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 1254–1259. https://doi.org/10.1.1.111.7144

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., i Chen, Z. (2020). *GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding* (No. arXiv:2006.16668). arXiv. http://arxiv.org/abs/2006.16668

Leroy, G., i Rindflesch, T. C. (2005). Effects of information and machine learning algorithms on word sense disambiguation with small datasets. *International Journal of Medical Informatics*, *74*(7–8), 573–585. https://doi.org/10/b5mjz7

Lesk, M. (1977). *sif—A tool for comparing and merging files* [Software].

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, *10*(8), 707–710.

Levy, A., Shalom, B. R., i Chalamish, M. (2024). *A Guide to Similarity Measures*. https://ar5iv.labs.arxiv.org/html/2408.07706

Li, M., Chen, X., Li, X., Ma, B., i Vitanyi, P. M. B. (2004). The Similarity Metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264. https://doi.org/10/d5r5jp

Li, Z., Jiang, X., Shang, L., i Li, H. (2018). *Paraphrase Generation with Deep Reinforcement Learning*. 3865–3878. https://doi.org/10/ghjqh6

Library Learning Space. (2019, ožujak 7). iParadigms acquires Ephorus to support international expansion. *Access*. https://librarylearningspace.com/iparadigms-acquires-ephorus-support-international-expansion/

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., i Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (No. arXiv:1907.11692). arXiv. https://doi.org/10.48550/arXiv.1907.11692

Luan, Y., Eisenstein, J., Toutanova, K., i Collins, M. (2021). Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*, *9*, 329–345. https://doi.org/10.1162/tacl_a_00369

Lukashenko, R., Graudina, V., i Grundspenkis, J. (2007). *Computer-based plagiarism detection methods and tools: An overview*. 40. https://doi.org/10/bjjp87

Luu, V.-T., Forestier, G., Weber, J., Bourgeois, P., Djelil, F., i Muller, P.-A. (2020). A review of alignment based similarity measures for web usage mining. *Artificial Intelligence Review*, *53*. https://doi.org/10/gjvnj5

Madnani, N., Tetreault, J., i Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. *Proceedings of the 2012 conference of the north*

american chapter of the association for computational linguistics: Human language technologies, 182–190. https://aclanthology.org/N12-1019.pdf

Mahmoud, A., i Zrigui, M. (2019). Sentence Embedding and Convolutional Neural Network for Semantic Textual Similarity Detection in Arabic Language. *Arabian Journal for Science and Engineering*, *44*(11), 9263–9274. https://doi.org/10/ghjqjj

Manning, C. D., Raghavan, P., i Schutze, H. (2009). An Introduction to Information Retrieval. *Online*, *c*, 569. https://doi.org/10/bh25gw

Marsi, E., i Krahmer, E. (2010). Automatic analysis of semantic similarity in comparable text through syntactic tree matching. *Proceedings of the 23rd International Conference on Computational Linguistics*, 752–760. http://dl.acm.org/citation.cfm?id=1873866

Marsi, E., i Krahmer, E. (2013). Automatic Tree Matching for Analysing Semantic Similarity in Comparable Text. U P. Spyns i J. Odijk (Ur.), *Essential Speech and Language Technology for Dutch: Results by the STEVIN programme* (str. 129–145). Springer. https://doi.org/10.1007/978-3-642-30910-6_8

Martin, B. (1994). Plagiarism: A misplaced emphasis. *Journal of Information Ethics*, *3*(2), 36–47.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, *405*(2), 442–451. https://doi.org/10.1016/0005-2795(75)90109-9

Maurer, H. A., Kappe, F., i Zaka, B. (2006). Plagiarism—A Survey. *Journal of Universal Computer Science*, *12*(8), 1050–1084.

McCulloch, W. S., i Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

McLachlan, G. J. (1999). Mahalanobis Distance. *Resonance*, *4*, 20–26. https://doi.org/10.1007/BF02834632

Ménard, J. (2021, ožujak 28). Turnitin Acquiring Ouriginal: What's left on the market. *ListEdTech*. https://listedtech.com/blog/turnitin-acquiring-ouriginal-whats-left-on-the-market/

Merriam-Webster Dictionary. (2016). *Definition of Plagiarism by Merriam-Webster*. http://www.merriam-webster.com/dictionary/plagiarism

Metatext. (2021). *NLP Hub—Metatext: Sentence transformers paraphrase distilroberta-base-v1 model*. https://metatext.io/models/sentence-transformers-paraphrase-distilroberta-base-v1

Meuschke, N., i Gipp, B. (2013). State-of-the-art in detecting academic plagiarism. *International Journal for Educational Integrity*, *9*(1), 50–71. https://doi.org/10/gj26t8

Mihalcea, R., Corley, C., i Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. *Proceedings of the 21st national conference on Artificial intelligence*, *6*, 775–780. https://www.aaai.org/Papers/AAAI/2006/AAAI06-123.pdf

Mikolov, T. (2012). *Statistical Language Models Based on Neural Networks*. *April*, 2015.

Mikolov, T., Chen, K., Corrado, G., i Dean, J. (2013a). Distributed Representations of Words

and Phrases and their Compositionality. *Nips*, 1–9. https://doi.org/10/c7rpzm

Mikolov, T., Chen, K., Corrado, G., i Dean, J. (2013b). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. https://arxiv.org/abs/1301.3781

Mikolov, T., Kombrink, S., Deoras, A., Burget, L., i Černocký, J. (2011). RNNLM --- Recurrent Neural Network Language Modeling Toolkit. *Proceedings of ASRU 2011*, 1–4.

Mikolov, T., i Zweig, G. (2012). Context dependent recurrent neural network language model. *2012 IEEE Spoken Language Technology Workshop (SLT)*, 234–239. https://doi.org/10/ghjx7k

Miller, W., i Myers, E. W. (1985). A file comparison program. *Software: Practice and Experience*, *15*(11), 1025–1040. https://doi.org/10.1002/spe.4380151102

Nalisnick, E., i Ravi, S. (2017). *Learning the Dimensionality of Word Embeddings* (No. arXiv:1511.05392). arXiv. http://arxiv.org/abs/1511.05392

Namrata. (2020, srpanj 27). *PlagScan and Urkund—At the top for detecting plagiarism*. https://www.ouriginal.com/plagscan-and-urkund-ranked-as-top-software/

Nandakumar, V., Mi, P., i Liu, T. (2023). *Why can neural language models solve next-word prediction? A mathematical perspective* (No. arXiv:2306.17184). arXiv. https://doi.org/10.48550/arXiv.2306.17184

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., i Mian, A. (2024). *A Comprehensive Overview of Large Language Models* (No. arXiv:2307.06435). arXiv. http://arxiv.org/abs/2307.06435

Oberreuter, G., i Velásquez, J. D. (2013). Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications*, *40*(9), 3756–3763. https://doi.org/10/gcpf36

OpenAI. (2024a). *ChatGPT [Large language model]*. https://chatgpt.com

OpenAI. (2024b, listopad 23). *OpenAI Platform: Vector embeddings*. https://platform.openai.com

Oxford Dictionary. (2018). *Definition of plagiarism in English*. http://www.oxforddictionaries.com/definition/english/plagiarism

Park, C. (2004). Rebels without a clause: Towards an institutional framework for dealing with plagiarism by students. *Journal of Further and Higher Education*, *28*(3), 291–306. https://doi.org/10/cbrg4w

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., i Lerer, A. (2017). *Automatic differentiation in PyTorch*. https://openreview.net/forum?id=BJJsrmfCZ

Patwardhan, S., Banerjee, S., i Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. U *Computational linguistics and intelligent text processing* (str. 241–257). Springer.

Pavlić, M., Jakupović, A., i Meštrović, A. (2013). Nodes of Knowledge Method for Knowledge Representation. *Informatologia*, *46*(3), 206.

Pavlić, M., Meštrović, A., i Jakupović, A. (2013). Graph-Based Formalisms for Knowledge

Representation. *Proceedings of the 17th World Multi-Conference on Systemics Cybernetics and Informatics (WMSCI 2013)*, *2*, 200–204.

Pejaković, A. (2009). *Predviđanje mjesta proteinskih interakcija iz slijeda aminokiselinskih ostataka i relativne površine dostupne otapalu*. University of Zagreb.

Pennington, J., Socher, R., i Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10/gfshwg

Perone, C. S. (2013, rujan). *Machine Learning: Cosine Similarity for Vector Space Models (Part III) | Terra Incognita*. http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

Perry, J. W., i Kent, A. (1958). *Tools for Machine Literature Searching: Semantic Code Dictionary, Equipment, Procedures*. Interscience Publishers.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., i Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10/gft5gf

Pinaya, W. H. L., Graham, M. S., Kerfoot, E., Tudosiu, P.-D., Dafflon, J., Fernandez, V., Sanchez, P., Wolleb, J., Costa, P. F. da, Patel, A., Chung, H., Zhao, C., Peng, W., Liu, Z., Mei, X., Lucena, O., Ye, J. C., Tsaftaris, S. A., Dogra, P., … Cardoso, M. J. (2023). *Generative AI for Medical Imaging: Extending the MONAI Framework* (No. arXiv:2307.15208). arXiv. http://arxiv.org/abs/2307.15208

Plagiarism.org. (2017). *What is Plagiarism?* https://www.plagiarism.org/article/what-is-plagiarism

Princeton University. (2010). *About WordNet*. https://wordnet.princeton.edu/

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., i Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning*, 8748–8763. https://proceedings.mlr.press/v139/radford21a.html

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., i Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*. *1*(8), 24.

Rajagopal, D., Cambria, E., Olsher, D., i Kwok, K. (2013). A graph-based approach to commonsense concept extraction and semantic similarity detection. *Proceedings of the 22nd international conference on World Wide Web companion*, 565–570. https://doi.org/10/gj26vb

Ram, R. V. S. V. S., Stamatatos, E., i Devi, S. L. L. (2014). Identification of Plagiarism Using Syntactic and Semantic Filters. U *Computational Linguistics and Intelligent Text Processing* (str. 495–506). Springer.

Ramaprabha, J., Das, S., i Mukerjee, P. (2018). Survey on sentence similarity evaluation using deep learning. *Journal of Physics: Conference Series*, *1000*(1), 012070. https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012070/meta

Rana, D. S., i Mishra, P. K. (2019). Paraphrase Detection using Dependency Tree Recursive Autoencoder. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 678–683. https://ieeexplore.ieee.org/abstract/document/8728539/?casa_token=3fevG5Q-WWIAAAAA:NR_lAxWMdJTYwV-mBE3lKbyer7BYMYYLX9fprtWyZu6QWYeHk6OnosXYcqfxwXs91i2ZGuZnTlEs

Reimers, N. (2021). *SentenceTransformers Documentation*. https://www.sbert.net/

Reimers, N., i Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. http://arxiv.org/abs/1908.10084

Roig, M. (2006). Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing. *St John's University*.

Rumelhart, D. E., Hinton, G. E., i Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Rus, V., Lintean, M. C. C., Banjade, R., Niraula, N. B. B., i Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. *ACL (Conference System Demonstrations)*, 163–168.

Saha, R., i Kumar, G. B. (2021). A Novel Approach for Developing Paraphrase Detection System using Machine Learning. *International Journal of Computer Applications*, *975*, 8887. https://www.researchgate.net/profile/Rudradityo-Saha/publication/352658644_A_Novel_Approach_for_Developing_Paraphrase_Detection_System_using_Machine_Learning/links/60d57ddd299bf1ea9ebae746/A-Novel-Approach-for-Developing-Paraphrase-Detection-System-using-Machine-Learning.pdf

Sahlgren, M. (2005). An introduction to random indexing. *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering*.

Salton, G., i Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10/bf8x8m

Salton, G., Wong, A., i Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10/fw8vv8

Samuelson, P. (1994). Self-plagiarism or fair use. *Communications of the ACM*, *37*(8), 21–25.

Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., Stamatatos, E., i Villaseñor-Pineda, L. (2019). Paraphrase plagiarism identification with character-level features. *Pattern Analysis and Applications*, *22*(2), 669–681. https://doi.org/10.1007/s10044-017-0674-z

Sanh, V., Debut, L., Chaumond, J., i Wolf, T. (2020). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter* (No. arXiv:1910.01108). arXiv. https://doi.org/10.48550/arXiv.1910.01108

Saqaabi, A. A., Akrida, E., Cristea, A., i Stewart, C. (2022). A Paraphrase Identification

Approach in Paragraph Length Texts. *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, 358–367. https://doi.org/10.1109/ICDMW58026.2022.00055

Sasaki, Y. (2007). The truth of the F-measure. *Teach tutor mater*, *1*(5), 1–5. https://nicolasshu.com/assets/pdf/Sasaki_2007_The%20Truth%20of%20the%20F-measure.pdf

Schleimer, S., Wilkerson, D. S. S., i Aiken, A. (2003). Winnowing: Local algorithms for document fingerprinting. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 76–85. https://doi.org/10/cqm2dc

Schwarzenegger, C., i Wohlers, W. (2006). Quellen zitieren, nicht plagiieren. *Universität Zürich Unijournal 4/06*, 16.

scikit-learn developers. (2021). *3.3. Metrics and scoring: Quantifying the quality of predictions*. Scikit-Learn. https://scikit-learn.org/stable/modules/model_evaluation.html

Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., Kannan, A., Sainath, T., Cao, Y., Chiu, C.-C., He, Y., Chorowski, J., Hinsu, S., Laurenzo, S., Qin, J., Firat, O., Macherey, W., Gupta, S., Bapna, A., … Rondon, P. (2019). *Lingvo: A Modular and Scalable Framework for Sequence-to-Sequence Modeling* (No. arXiv:1902.08295). arXiv. http://arxiv.org/abs/1902.08295

Shuang, K., Zhang, Z., Loo, J., i Su, S. (2020). Convolution–deconvolution word embedding: An end-to-end multi-prototype fusion embedding method for natural language processing. *Information Fusion*, *53*(June 2019), 112–122. https://doi.org/10/ghjqhd

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., i Pinto, D. (2014). Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, *18*(3), Article 3. https://doi.org/10/gcpzhs

Socher, R. (2014). *Recursive deep learning for natural language processing and computer vision*. Stanford University.

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., i Manning, C. D. (2011). Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. *Advances in neural information processing systems*, 801–809.

Sundaresan, S. (2022, prosinac 8). Understanding Large Language Models. *Medium*. https://medium.com/@artofsaience/understanding-large-language-models-6664be71988e

Swets, J. A. (Ur.). (1964). *Signal Detection and Recognition by Human Observers: Contemporary Readings* (First Edition). John Wiley & Sons.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J., i Bašić, B. D. (2012). Takelab: Systems for Measuring Semantic Text Similarity. *First Joint Conference on Lexical and Computational Semantics (*SEM)*, 441–448. http://dl.acm.org/citation.cfm?id=2387708

Taloni, A., Scorcia, V., i Giannaccare, G. (2023). Modern threats in academia: Evaluating plagiarism and artificial intelligence detection scores of ChatGPT. *Eye*, *38*(2), 397–400. https://doi.org/10.1038/s41433-023-02678-7

Tan, P.-N., Steinbach, M., i Kumar, V. (2014). *Introduction to data mining* (New internat. edition). Pearson.

Teh, Y., Jordan, M., Beal, M., i Blei, D. (2006). Hierarchical Dirichlet Processes. *Machine Learning*, *101*(476), 1–30. https://doi.org/10/bgxctd

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., i Pavlick, E. (2019). What do you learn from context? Probing for sentence structure in contextualized word representations. *7th International Conference on Learning Representations, ICLR 2019*, 1–17. http://arxiv.org/abs/1905.06316

The Linux Foundation. (2023). *PyTorch documentation*. https://pytorch.org/docs/stable/index.html

Thompson, V., i Bowerman, C. (2017). Methods for Detecting Paraphrase Plagiarism. *CoRR*, *abs/1712.1*, 1–21. http://arxiv.org/abs/1712.10309

Turing.com. (2024). *Guide to deciding the perfect distance metric for your ML model.* https://www.turing.com/kb/how-to-decide-perfect-distance-metric-for-machine-learning-model

Turney, P. (2001). *Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL*. https://doi.org/10/bjpxwq

Turney, P. D., i Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188. https://doi.org/10/gd85zk

Turnitin. (2021). *Turnitin acquires Ouriginal*. https://www.turnitin.com/press/turnitin-acquires-ouriginal

Turnitin. (2023). *Generative AI in Higher Education: Fall 2023 Update of Time For Class Study.* https://go.turnitin.com/l/45292/2023-11-15/cl2c7v/45292/1700061088dRwB8WtL/GenAI_IN_HIGHER_EDUCATION_FALL_2023_UPDATE_TIME_FOR_CLASS_STUDY.pdf

Turnitin. (2024). *About Turnitin*. https://www.turnitin.com/about/

Turnitin AI Technical Staff. (2023). *White Paper: Turnitin's AI writing detection model architecture and testing protocol*. Turnitin LLC. https://go.turnitin.com/l/45292/2023-11-21/cl3clv/45292/170057442751hShVzB/TII_AI_HE_AIWritingDetectionModel_Whitepaper_US_0923.pdf?utm_source=pardot&utm_medium=email&utm_content=whitepaper&utm_campaign=102356

Turnitin Europe. (2016). *Plagiarism in a Digital World series*. Turnitin. http://go.turnitin.com/en/whitepaperLP1

Ushio, A., i Liberatore, F. (2024). *Back to the Basics: A Quantitative Analysis of Statistical and Graph-Based Term Weighting Schemes for Keyword Extraction*. https://ar5iv.labs.arxiv.org/html/2104.08028

Van Rijsbergen, C. (1979). Information retrieval: Theory and practice. *Proceedings of the joint IBM/University of Newcastle upon tyne seminar on data base systems*, *79*, 1–14.

http://homepages.cs.ncl.ac.uk/brian.randell/Seminars/146.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., i Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*. https://proceedings.neurips.cc/paper/7181-attention-is-all

Vihar Kurama. (2024). *PyTorch vs. TensorFlow for Deep Learning*. https://builtin.com/data-science/pytorch-vs-tensorflow

Voorhees, E. (1993). Using WordNet to Disambiguate Word Senses for Text Retrieval. U *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Infofmation Retrieval* (str. 180). https://doi.org/10.1145/160688.160715

Voxco. (2024). *Matthews's correlation coefficient: Definition, Formula and advantages - Voxco*. https://www.voxco.com/blog/matthewss-correlation-coefficient-definition-formula-and-advantages/

Vrbanec, T., i Meštrović, A. (2017). The struggle with academic plagiarism: Approaches based on semantic similarity. *40th International Convention on Information and Communication Technology, Electronics and Microelectronics*, 976–981. https://doi.org/10/gj26vx

Vrbanec, T., i Meštrović, A. (2020). Corpus-Based Paraphrase Detection Experiments and Review. *Information*, *11*(5), 241. https://doi.org/10/ghjtff

Vrbanec, T., i Meštrović, A. (2021a). Relevance of Similarity Measures Usage for Paraphrase Detection: *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 129–138. https://doi.org/10/gndtr5

Vrbanec, T., i Meštrović, A. (2021b). Taxonomy of academic plagiarism methods. *Journal of the Polytechnic of Rijeka*, *9*(1), 283–300. https://doi.org/10.31784/zvr.9.1.17

Vrbanec, T., i Meštrović, A. (2023). Comparison study of unsupervised paraphrase detection: Deep learning—The key for semantic similarity detection. *Expert Systems*, *40*(9), e13386. https://doi.org/10.1111/exsy.13386

Všianský, R. (2019). *Scored Greedy String Tilling*. GitHub. https://github.com/rvsia/scored-greedy-string-tiling

Wahle, J. P., Gipp, B., i Ruas, T. (2023). *Paraphrase Types for Generation and Detection* (No. arXiv:2310.14863). arXiv. https://doi.org/10.48550/arXiv.2310.14863

Wang, J., i Dong, Y. (2020). Measurement of Text Similarity: A Survey. *Information*, *11*(9), Article 9. https://doi.org/10.3390/info11090421

Williams, J. B. (2005). *Plagiarism: Deterrence, Detection and Prevention* (str. 20). Universitas 21 Global.

Williams, T. (2023, veljača 14). *Turnitin announces AI detector with '97 per cent accuracy'*. Times Higher Education (THE). https://www.timeshighereducation.com/news/turnitin-announces-ai-detector-97-cent-accuracy

World Intellectual Property Organization (WIPO). (1886). *Summary of the Berne Convention for the Protection of Literary and Artistic Works (1886)*. http://www.wipo.int/treaties/en/ip/berne/summary_berne.html

Wu, W., Wang, H., Liu, T., Ma, S., i Key, M. (2018). Phrase-level Self-Attention Networks for Universal Sentence Encoding. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3729–3738. https://doi.org/10/ggv87s

Wu, Z., i Palmer, M. (1994). Verbs semantics and lexical selection. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* -, 133–138. https://doi.org/10/dvrwwn

Xu, Z., Jain, S., i Kankanhalli, M. (2024). *Hallucination is Inevitable: An Innate Limitation of Large Language Models* (No. arXiv:2401.11817). arXiv. https://doi.org/10.48550/arXiv.2401.11817

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., i Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in neural information processing systems*, *NeurIPS*, 5754–5764. http://arxiv.org/abs/1906.08237

Yeh, K.-C., Chi, J.-A., Lian, D.-C., i Hsieh, S.-K. (2023). Evaluating Interfaced LLM Bias. U J.-L. Wu i M.-H. Su (Ur.), *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)* (str. 292–299). The Association for Computational Linguistics and Chinese Language Processing (ACLCLP). https://aclanthology.org/2023.rocling-1.37

Yin, W., i Schuetze, H. (2015). Convolutional Neural Network for Paraphrase Identification. *Naacl-2015*, 901–911. https://doi.org/10/ghjqhw

Zablocki, E., Piwowarski, B., Soulier, L., i Gallinari, P. (2018). Learning multi-modal word representation grounded in visual context. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) Learning*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16113

Zervanou, K., Iosif, E., i Potamianos, A. (2014). Word Semantic Similarity for Morphologically Rich Languages. *LREC*, 1642–1648.

Zhang, A., Lipton, Z. C., Li, M., i Smola, A. J. (2023). *Dive into Deep Learning* (No. arXiv:2106.11342). arXiv. https://doi.org/10.48550/arXiv.2106.11342

Zhou, C., Qiu, C., i Acuna, D. E. (2022). *Paraphrase Identification with Deep Learning: A Review of Datasets and Methods* (No. arXiv:2212.06933). arXiv. http://arxiv.org/abs/2212.06933

Zhou, J., Liu, G., i Sun, H. (2018). Paraphrase Identification Based on Weighted URAE, Unit Similarity and Context Correlation Feature. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11109 LNAI*, 41–53. https://doi.org/10/ghjqjh

Zu Eissen, S. M. M., i Stein, B. (2006). Intrinsic plagiarism detection. U *Advances in Information Retrieval* (str. 565–569). Springer. http://link.springer.com/10.1007%2F11735106_66

Dodatak: Kratice

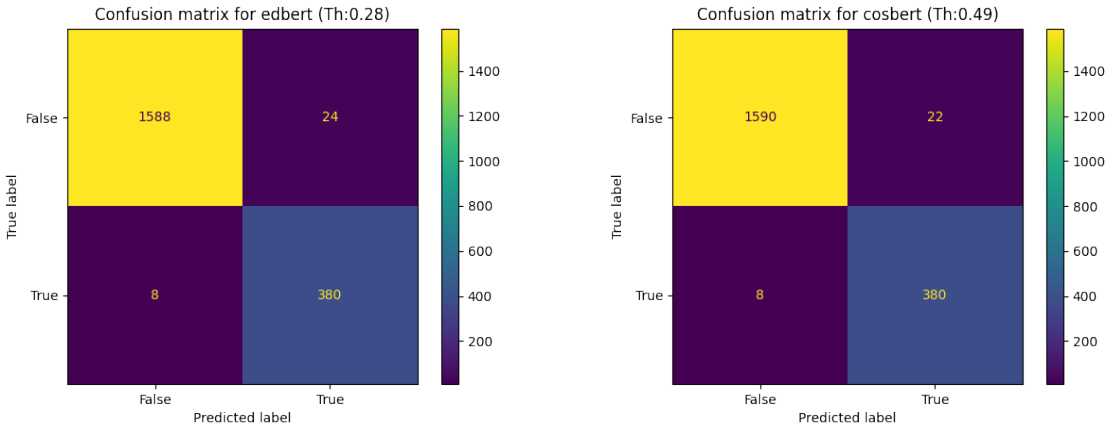
MCC = Matthews Correlation Coefficient
cosbert = Cosine Similarity (BERT)
cosd2vddbow = Cosine Similarity (Doc2Vec Documents DBoW)
cosd2vddm = Cosine Similarity (Doc2Vec Documents DM)
cosd2vwdbow = Cosine Similarity (Doc2Vec Words DBoW)
cosd2vwdm = Cosine Similarity (Doc2Vec Words DM)
coselmo = Cosine Similarity (ELMo)
cosft = Cosine Similarity (FastText)
cosgld = Cosine Similarity (GloVe Documents)
cosglw = Cosine Similarity (GloVe Words)
coslaser = Cosine Similarity (Laser Embeddings)
cosuse = Cosine Similarity (USE)
cosw2vcbow = Cosine Similarity (Word2Vec CBoW)
cosw2vsg = Cosine Similarity (Word2Vec Skip-Gram)
edbert = Euclidean Similarity (BERT)
edd2vddbow = Euclidean Similarity (Doc2Vec Documents DBoW)
edd2vddm = Euclidean Similarity (Doc2Vec Documents DM)
edd2vwdbow = Euclidean Similarity (Doc2Vec Words DBoW)
edd2vwdm = Euclidean Similarity (Doc2Vec Words DM)
edelmo = Euclidean Similarity (ELMo)
edft = Euclidean Similarity (FastText)
edgld = Euclidean Similarity (GloVe Documents)
edglw = Euclidean Similarity (GloVe Words)
edlaser = Euclidean Similarity (Laser Embeddings)
eduse = Euclidean Similarity (USE)
edw2vcbow = Euclidean Similarity (Word2Vec CBoW)
edw2vsg = Euclidean Similarity (Word2Vec Skip-Gram)
gwt = Greedy Word Tiling
hdp = Hierarchical Dirichlet Process Similarity
jacc = Jaccard Similarity
keep = Stop words were kept
lda = Latent Dirichlet Allocation Similarity
le = LogEntropy Similarity
lev = Levenshtein Similarity
lsi = Latent Semantic Indexing / Latent Semantic Analysis Similarity
mdbert = Manhattan Similarity (BERT)
mdd2vddbow = Manhattan Similarity (Doc2Vec Documents DBoW)
mdd2vddm = Manhattan Similarity (Doc2Vec Documents DM)
mdd2vwdbow = Manhattan Similarity (Doc2Vec Words DBoW)
mdd2vwdm = Manhattan Similarity (Doc2Vec Words DM)
mdelmo = Manhattan Similarity (ELMo)
mdft = Manhattan Similarity (FastText)
mdgld = Manhattan Similarity (GloVe Documents)

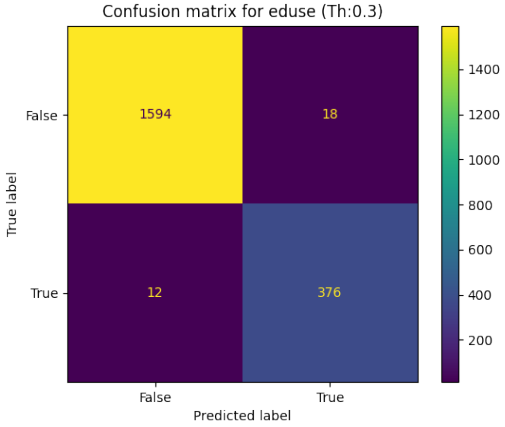
mdglw = Manhattan Similarity (GloVe Words)
mdlaser = Manhattan Similarity (Laser Embeddings)
mduse = Manhattan Similarity (USE)
mdw2vcbow = Manhattan Similarity (Word2Vec CBoW)
mdw2vsg = Manhattan Similarity (Word2Vec Skip-Gram)
rp = Random Projections / Random Indexing Similarity
scsd2vwdbow = Soft Cosine Similarity (Doc2Vec Words DBoW)
scsd2vwdm = Soft Cosine Similarity (Doc2Vec Words DM)
scsft = Soft Cosine Similarity (FastText)
scsglw = Soft Cosine Similarity (GloVe Words)
scsw2vcbow = Soft Cosine Similarity (Word2Vec CBoW)
scsw2vsg = Soft Cosine Similarity (Word2Vec Skip-Gram)
tfidf = Term Frequency – Inverse Document Frequency Similarity
wmdd2vwdbow = Word Mover's Similarity (Doc2Vec Words DBoW)
wmdd2vwdm = Word Mover's Similarity (Doc2Vec Words DM)
wmdft = Word Mover's Similarity (FastText)
wmdglw = Word Mover's Similarity (GloVe Words)
wmdw2vcbow = Word Mover's Similarity (Word2Vec CBoW)
wmdw2vsg = Word Mover's Similarity (Word2Vec Skip-Gram)

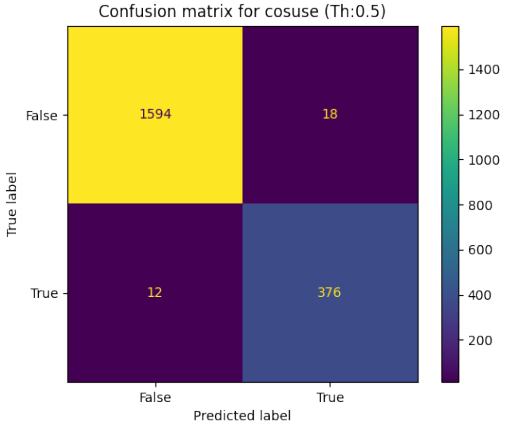
Dodatak: Matrice zabune za top 10 metoda prvog ciklusa (dokumenti)

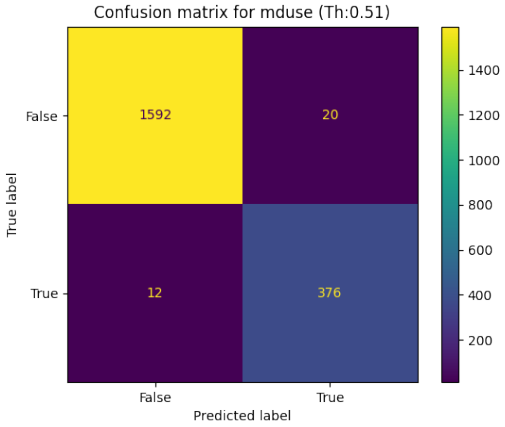
Matrica zabune pruža detaljniji uvid u performanse modela od točnosti (engl. *Accuracy*). Ona omogućuje izračun različitih metrika evaluacije kao što su preciznost, odziv i F1-mjeru. S obzirom da su za izradu matrice zabune potrebne vrijednosti *TP* (istinito pozitivno), *TN* (istinito negativno), *FP* (lažno pozitivno) i *FN* (lažno negativno), postoji problem njene izrade za korpuse kod kojih je proveden postupak unakrsne validacije jer priroda matrice zabune ne dopušta prosječne vrijednosti koje su uglavnom decimalne vrijednosti, već samo cijele brojeve. Za te korpuse (CS, VMEN), matrice zabune izrađene su zbrajanjem vrijednosti iz svih pet iteracija.

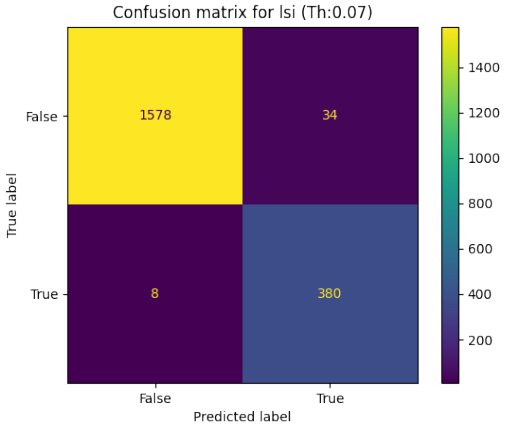
Tablica 24. Matrice zabune najboljih 10 metoda prvog ciklusa eksperimenata (CS)

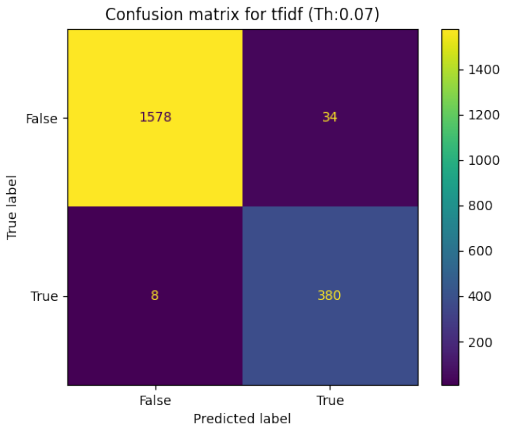


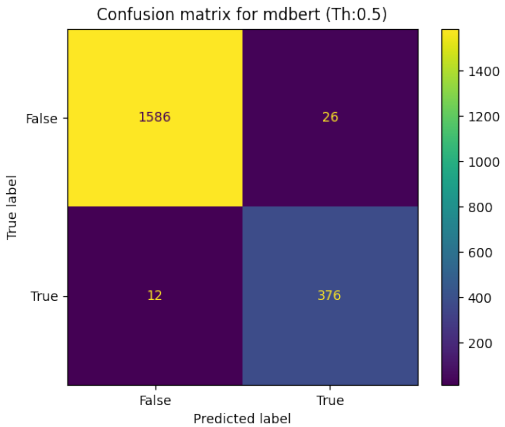

Confusion matrix for eduse (Th:0.3)
False
True
True label
1594
18
12
376
False
True
Predicted label
1400
1200
1000
800
600
400
200

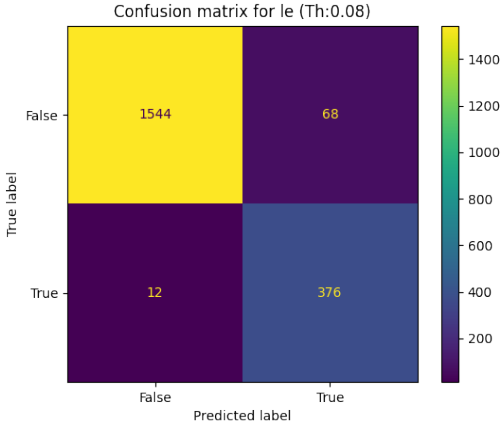

Confusion matrix for cosuse (Th:0.5)
False
True
True label
1594
18
12
376
False
True
Predicted label
1400
1200
1000
800
600
400
200

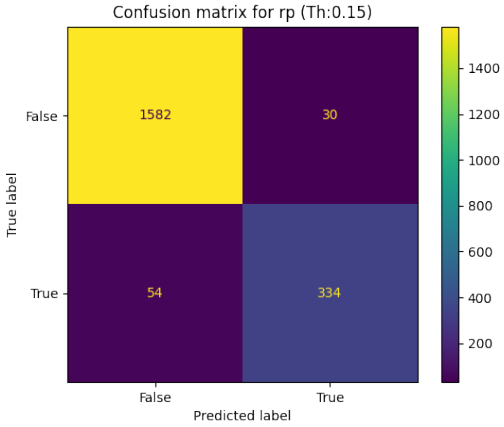

Confusion matrix for mduse (Th:0.51)
False
True
True label
1592
20
12
376
False
True
Predicted label
1400
1200
1000
800
600
400
200


Confusion matrix for lsi (Th:0.07)
False
True
True label
1578
34
8
380
False
True
Predicted label
1400
1200
1000
800
600
400
200

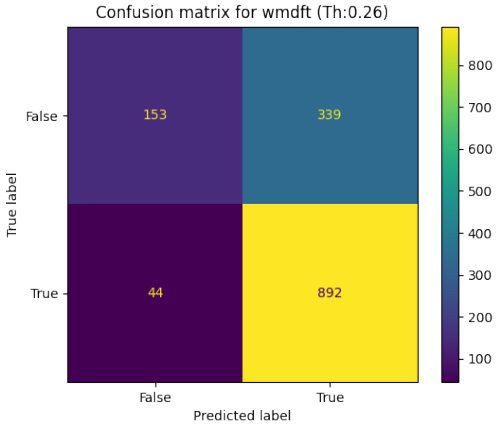

Confusion matrix for tfidf (Th:0.07)
False
True
True label
1578
34
8
380
False
True
Predicted label
1400
1200
1000
800
600
400
200

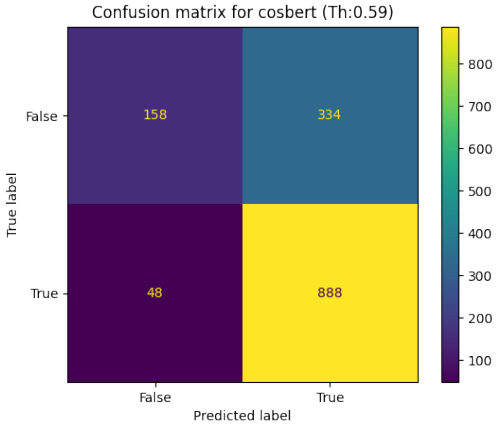

Confusion matrix for mdbert (Th:0.5)
False
True
True label
1586
26
12
376
False
True
Predicted label
1400
1200
1000
800
600
400
200

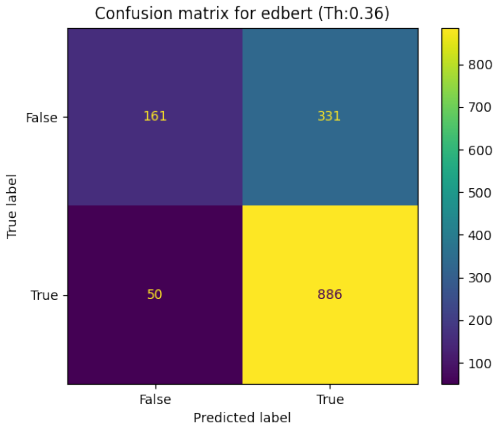


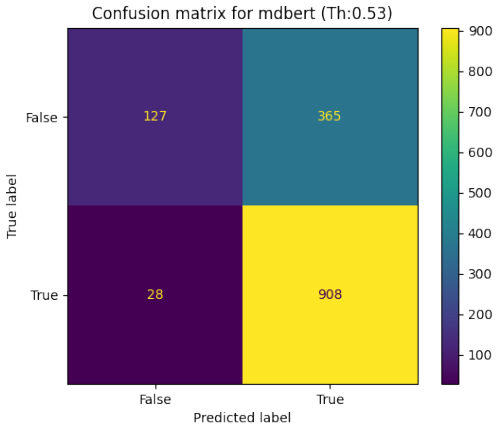


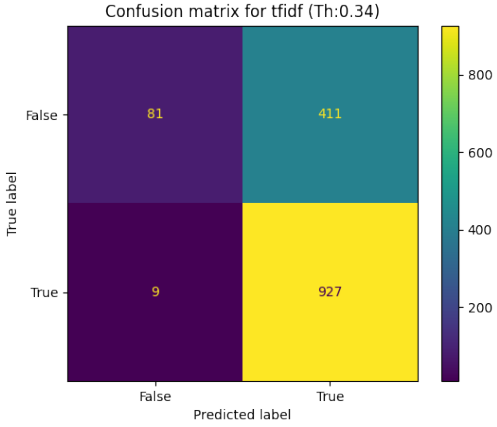
Tablica 25. Matrice zabune najboljih 10 metoda prvog ciklusa eksperimenata (MSRP)

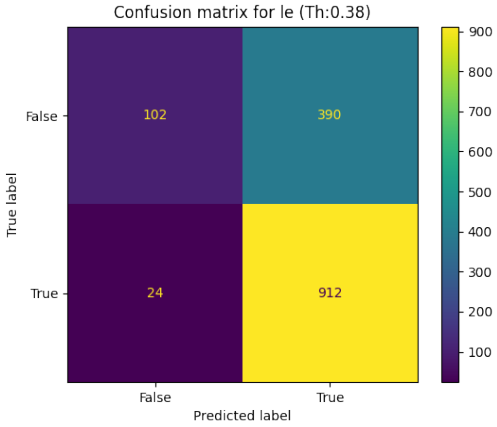


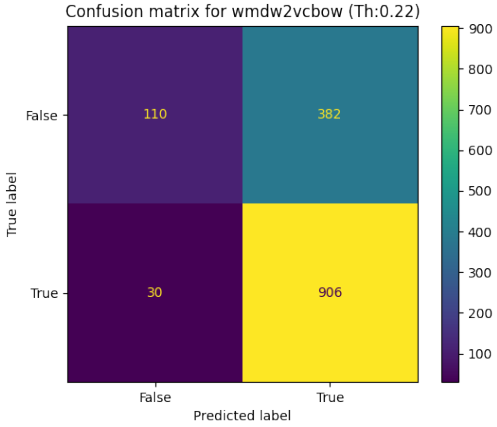


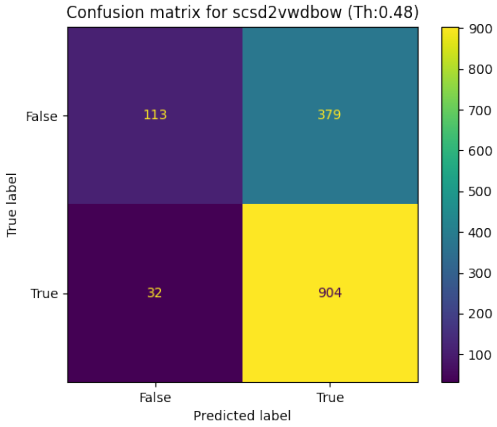


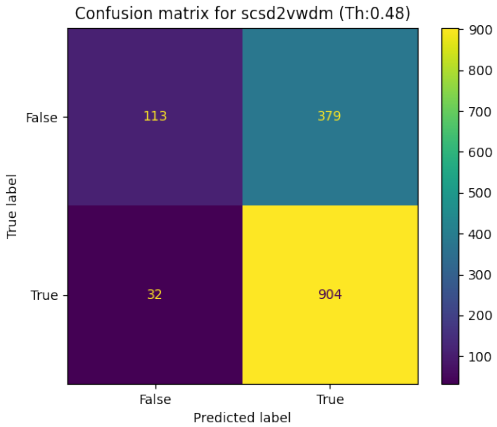


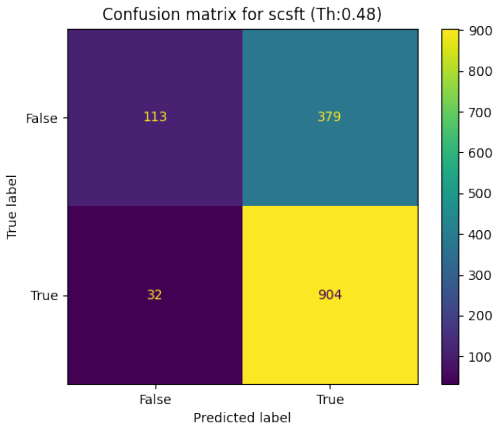

Confusion matrix for tfidf (Th:0.34)
True label
False
True
81
411
9
927
False
True
Predicted label
800
600
400
200


Confusion matrix for le (Th:0.38)
True label
False
True
102
390
24
912
False
True
Predicted label
900
800
700
600
500
400
300
200
100

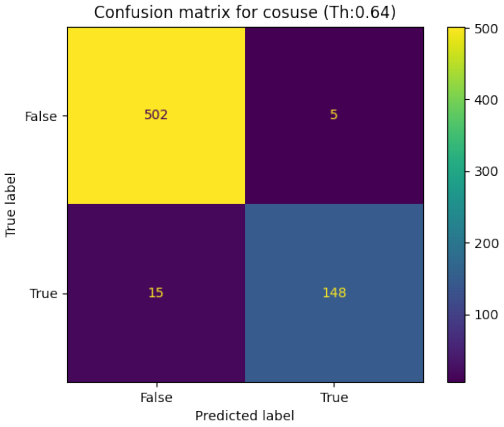

Confusion matrix for wmdw2vcbow (Th:0.22)
True label
False
True
110
382
30
906
False
True
Predicted label
900
800
700
600
500
400
300
200
100

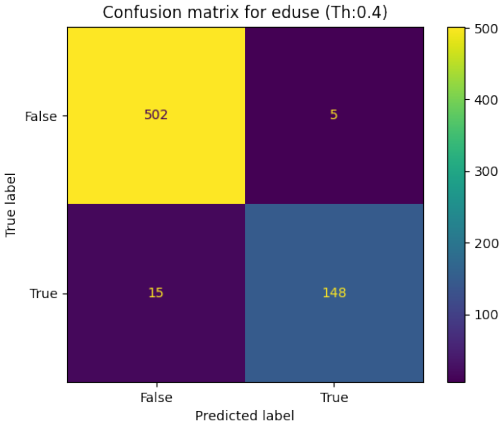

Confusion matrix for scsd2vwdbow (Th:0.48)
True label
False
True
113
379
32
904
False
True
Predicted label
900
800
700
600
500
400
300
200
100

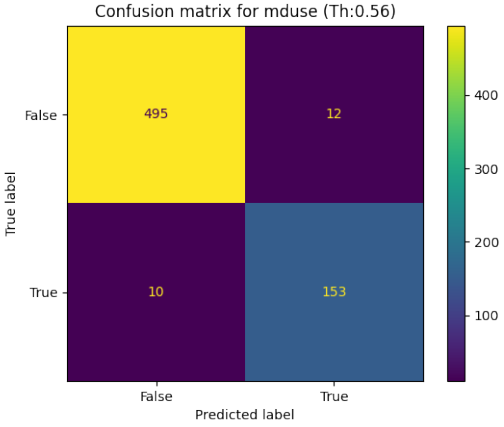

Confusion matrix for scsd2vwdm (Th:0.48)
True label
False
True
113
379
32
904
False
True
Predicted label
900
800
700
600
500
400
300
200
100

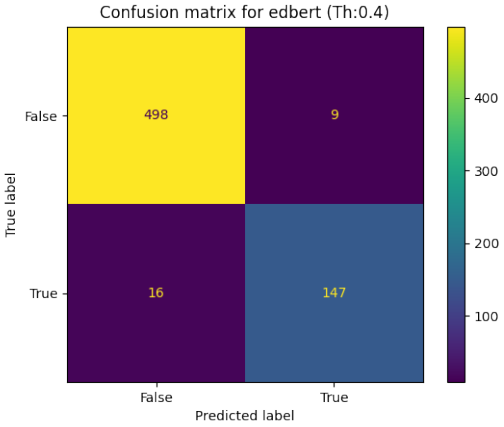

Confusion matrix for scsft (Th:0.48)
True label
False
True
113
379
32
904
False
True
Predicted label
900
800
700
600
500
400
300
200
100

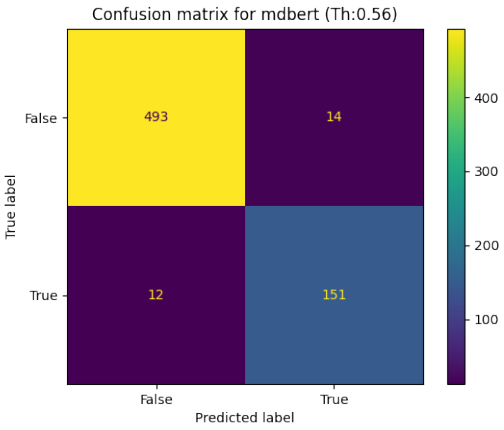
Tablica 26. Matrice zabune najboljih 10 metoda prvog ciklusa eksperimenata (P4PIN)

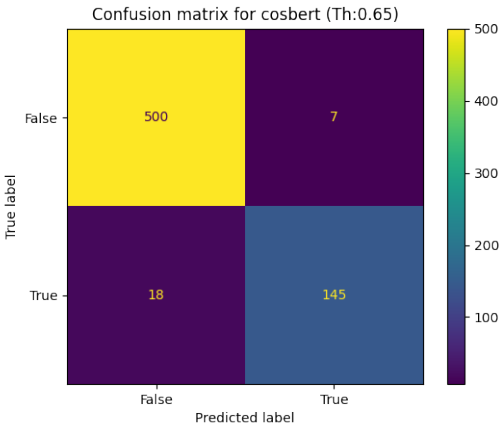


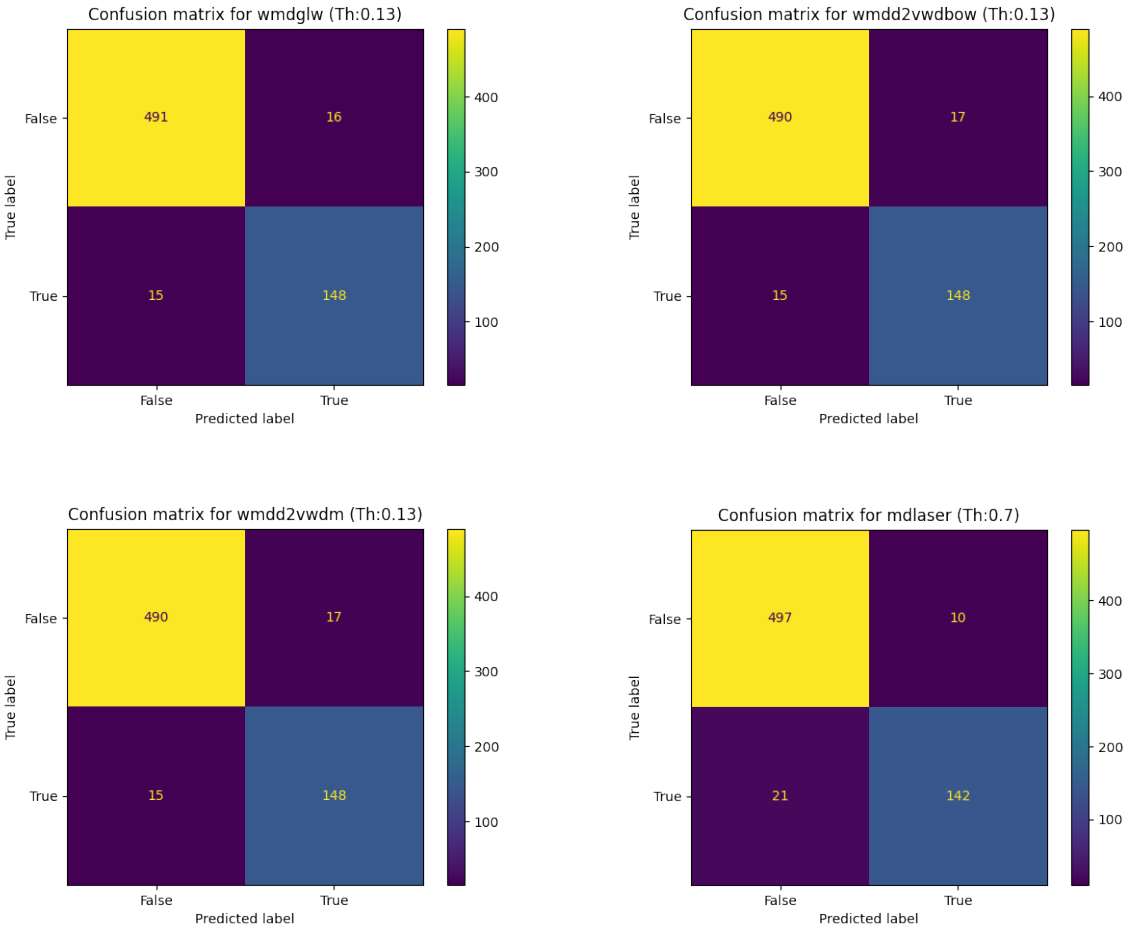




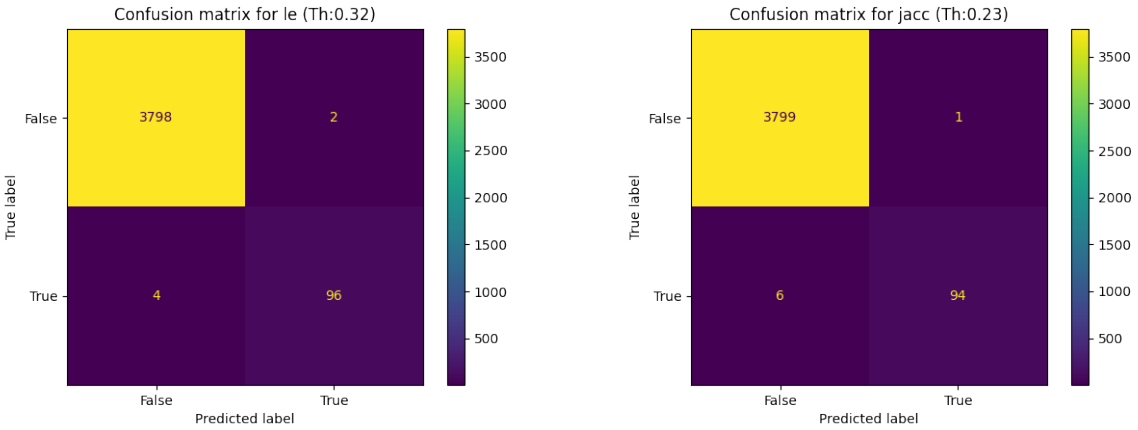


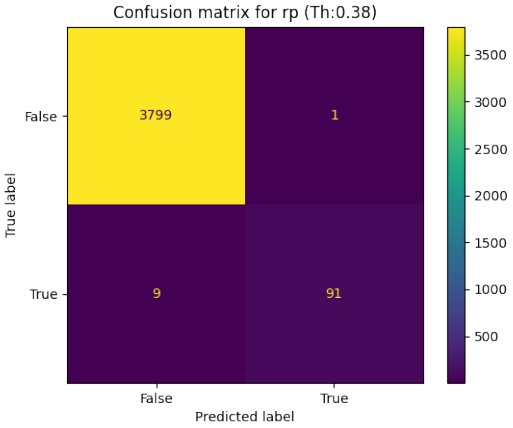


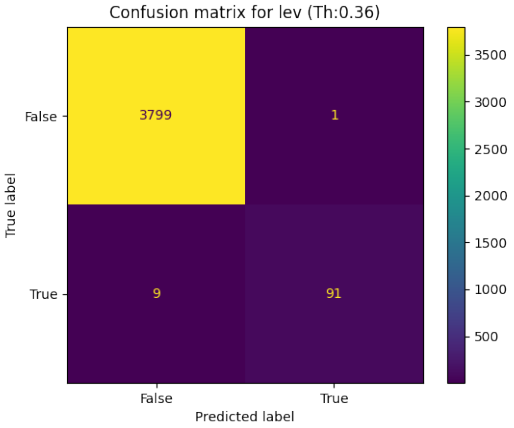


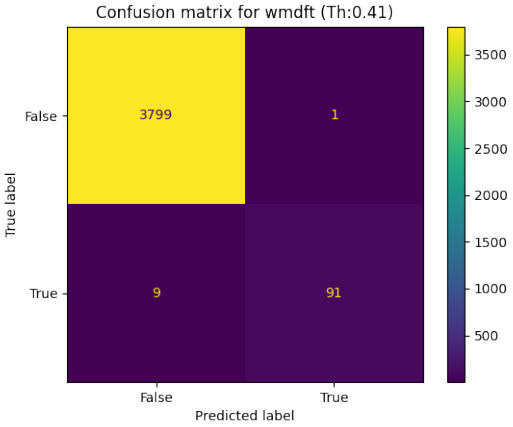


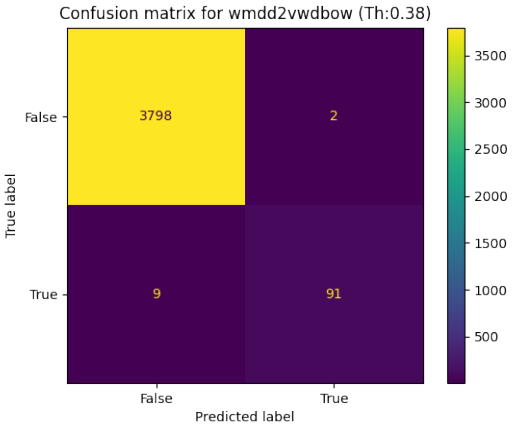
Tablica 27. Matrice zabune najboljih 10 metoda prvog ciklusa eksperimenata (VMEN)

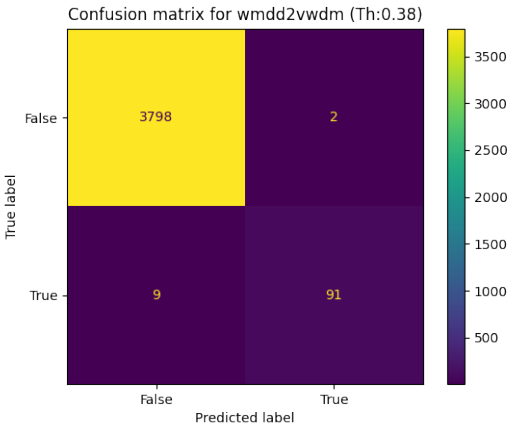


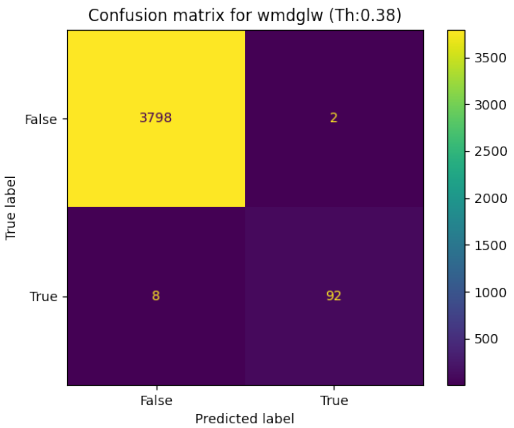

Confusion matrix for rp (Th:0.38)
True label
False
True
3799
1
9
91
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500

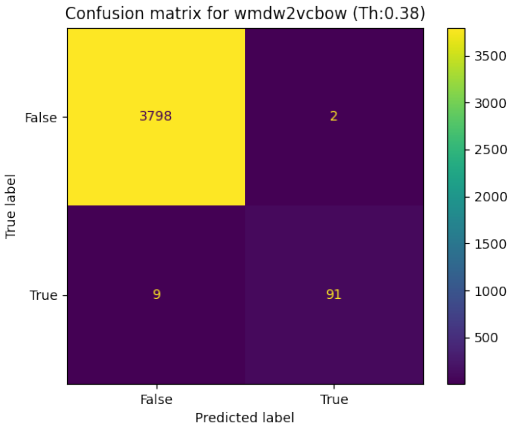

Confusion matrix for lev (Th:0.36)
True label
False
True
3799
1
9
91
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500

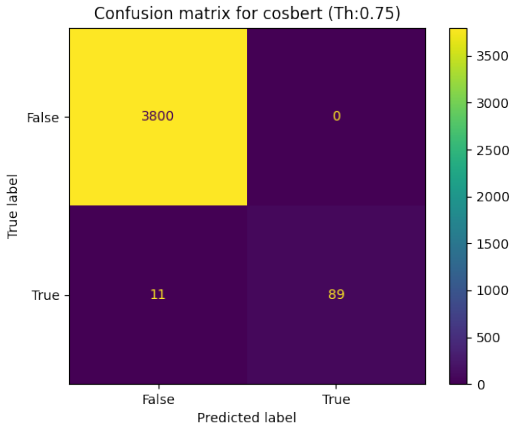

Confusion matrix for wmdft (Th:0.41)
True label
False
True
3799
1
9
91
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500


Confusion matrix for wmdd2vwdbow (Th:0.38)
True label
False
True
3798
2
9
91
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500

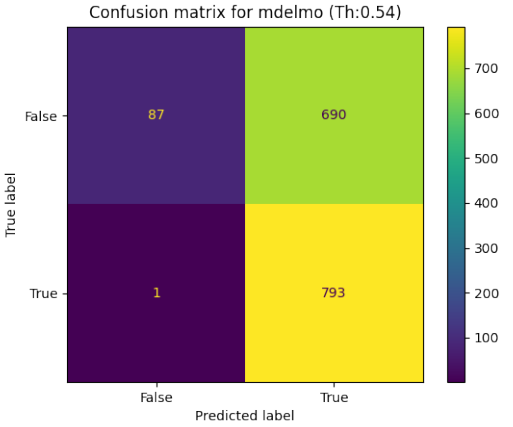

Confusion matrix for wmdd2vwdm (Th:0.38)
True label
False
True
3798
2
9
91
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500

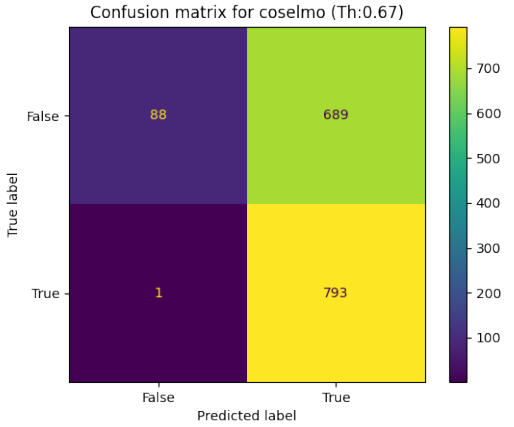

Confusion matrix for wmdglw (Th:0.38)
True label
False
True
3798
2
8
92
False
True
Predicted label
3500
3000
2500
2000
1500
1000
500

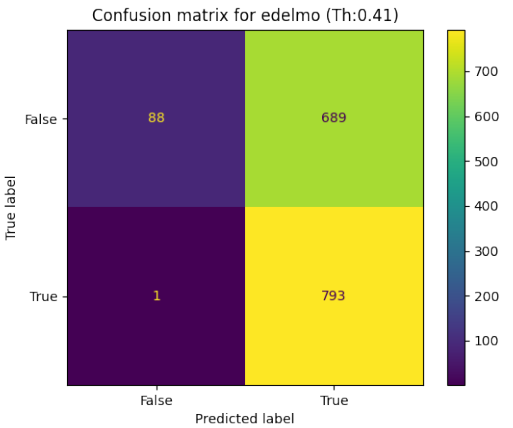


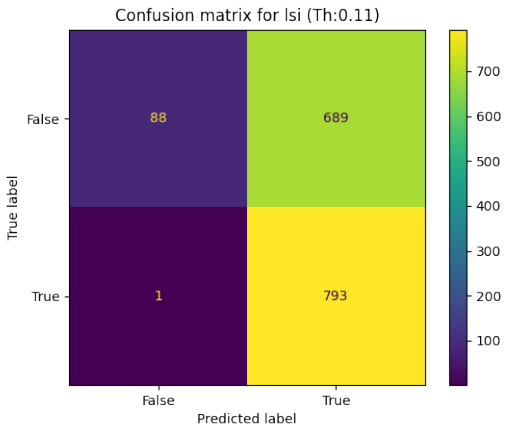


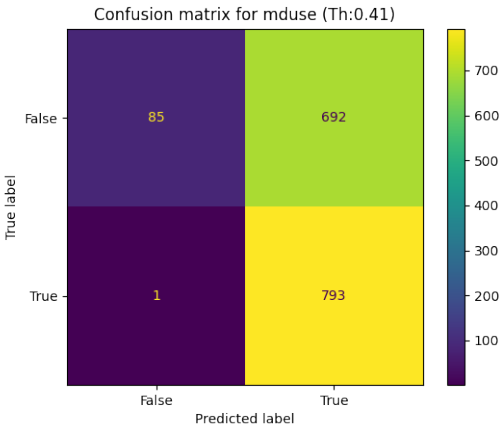
Tablica 28. Matrice zabune najboljih 10 metoda prvog ciklusa eksperimenata (Webis-11)

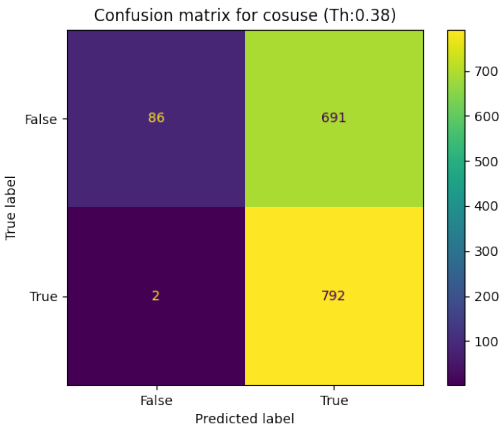


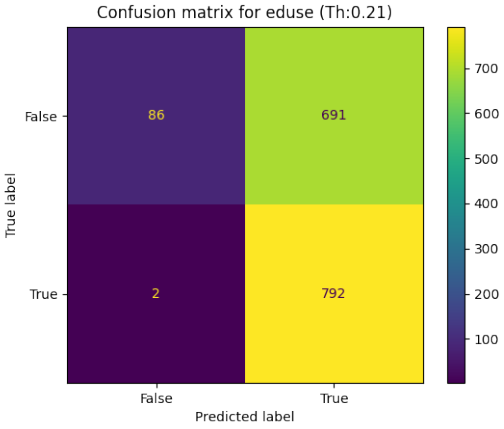


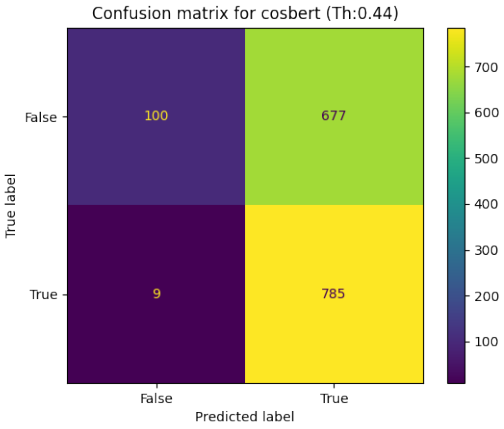


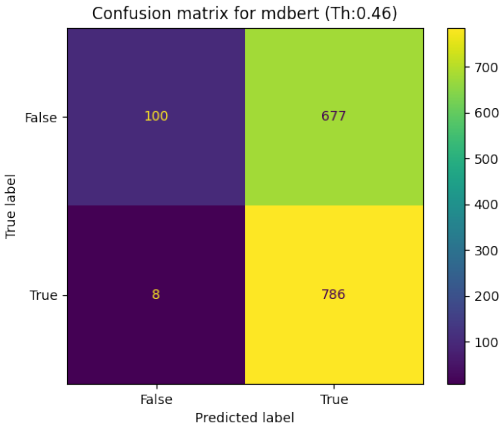


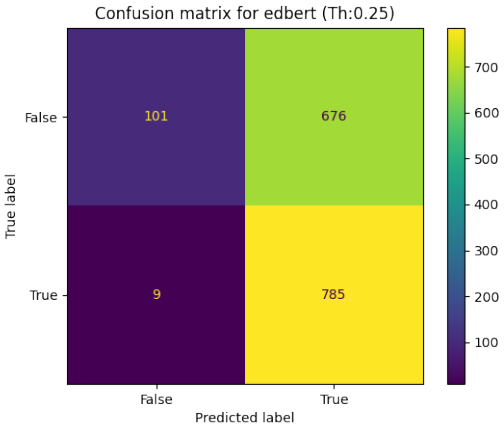

Confusion matrix for mduse (Th:0.41)
True label
False
True
85
692
1
793
False
True
Predicted label
700
600
500
400
300
200
100


Confusion matrix for cosuse (Th:0.38)
True label
False
True
86
691
2
792
False
True
Predicted label
700
600
500
400
300
200
100


Confusion matrix for eduse (Th:0.21)
True label
False
True
86
691
2
792
False
True
Predicted label
700
600
500
400
300
200
100


Confusion matrix for cosbert (Th:0.44)
True label
False
True
100
677
9
785
False
True
Predicted label
700
600
500
400
300
200
100


Confusion matrix for mdbert (Th:0.46)
True label
False
True
100
677
8
786
False
True
Predicted label
700
600
500
400
300
200
100


Confusion matrix for edbert (Th:0.25)
True label
False
True
101
676
9
785
False
True
Predicted label
700
600
500
400
300
200
100

Dodatak: Cjeloviti rezultati prvog ciklusa eksperimenata (dokumenti)

Ovaj dodatak sadrži tablice rezultata za pet korištenih korpusa kao izvora za sažete rezultate predstavljene u tablici 21. Naslovi tablica imaju skraćene naslove: *θ=Threshold* (granična vrijednost); *S=Specificity* (specifičnost); *A=Accuracy* (točnost); *P=Precision* (preciznost); *R=Recall* (Odziv); MCC=*Matthews Correlation Coefficient* (Matthewsov koeficijent korelacije).

Tablica 29. Rezultati prvog ciklusa eksperimenata za korpus CS

#	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	edbert	0.28	0.986	0.984	0.944	0.980	0.959	0.949	0.971	0.951
2	cosbert	0.49	0.987	0.984	0.948	0.975	0.959	0.952	0.968	0.951
3	eduse	0.30	0.988	0.984	0.955	0.970	0.959	0.956	0.964	0.952
4	cosuse	0.50	0.988	0.983	0.955	0.965	0.956	0.955	0.960	0.948
5	mduse	0.51	0.988	0.982	0.955	0.960	0.953	0.953	0.956	0.945
6	lsi	0.07	0.979	0.979	0.920	0.980	0.947	0.930	0.966	0.936
7	tfidf	0.07	0.979	0.979	0.920	0.980	0.947	0.930	0.966	0.936
8	mdbert	0.50	0.983	0.979	0.935	0.965	0.946	0.939	0.956	0.936
9	le	0.08	0.958	0.960	0.851	0.969	0.904	0.870	0.941	0.884
10	rp	0.15	0.977	0.952	0.906	0.853	0.873	0.891	0.860	0.848
11	jacc	0.13	0.968	0.934	0.854	0.793	0.822	0.840	0.804	0.783
12	mdelmo	0.70	0.953	0.919	0.813	0.785	0.789	0.801	0.784	0.747
13	coselmo	0.85	0.956	0.917	0.822	0.765	0.780	0.802	0.768	0.740
14	edelmo	0.61	0.952	0.915	0.813	0.770	0.778	0.796	0.769	0.737
15	wmdft	0.21	0.889	0.888	0.694	0.890	0.765	0.719	0.830	0.717
16	scsd2vwdbow	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703
17	scsd2vwdm	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703

#	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
18	scsft	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703
19	scsglw	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703
20	scsw2vcbow	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703
21	scsw2vsg	0.67	0.962	0.909	0.833	0.693	0.747	0.793	0.711	0.703
22	wmdw2vcbow	0.20	0.863	0.867	0.642	0.890	0.731	0.673	0.813	0.677
23	wmdw2vsg	0.20	0.863	0.867	0.642	0.890	0.731	0.673	0.813	0.677
24	wmdd2vwdbow	0.20	0.863	0.867	0.642	0.890	0.731	0.673	0.813	0.677
25	wmdd2vwdm	0.20	0.863	0.867	0.642	0.890	0.731	0.673	0.813	0.677
26	mdlaser	0.73	0.961	0.900	0.804	0.647	0.715	0.765	0.672	0.662
27	wmdglw	0.20	0.844	0.851	0.618	0.884	0.710	0.649	0.798	0.651
28	lev	0.28	0.917	0.879	0.696	0.719	0.702	0.697	0.711	0.631
29	gwt	0.02	0.958	0.892	0.795	0.616	0.688	0.746	0.642	0.636
30	cosd2vwdbow	0.72	0.979	0.894	0.868	0.545	0.664	0.770	0.586	0.632
31	edd2vwdbow	0.47	0.979	0.894	0.868	0.545	0.664	0.770	0.586	0.632
32	cosw2vcbow	0.73	0.982	0.895	0.881	0.534	0.661	0.776	0.578	0.633
33	cosd2vwdm	0.73	0.979	0.893	0.866	0.540	0.659	0.767	0.581	0.627
34	edd2vwdm	0.48	0.979	0.893	0.866	0.540	0.659	0.767	0.581	0.627
35	edw2vcbow	0.49	0.983	0.894	0.887	0.524	0.655	0.775	0.569	0.629
36	edlaser	0.64	0.929	0.870	0.688	0.626	0.652	0.672	0.635	0.576
37	coslaser	0.87	0.920	0.866	0.667	0.641	0.649	0.659	0.643	0.570
38	cosw2vsg	0.74	0.984	0.890	0.887	0.501	0.636	0.763	0.547	0.612
39	edw2vsg	0.50	0.988	0.891	0.909	0.491	0.633	0.771	0.539	0.617
40	cosglw	0.70	0.913	0.855	0.717	0.623	0.627	0.666	0.616	0.569
41	mdd2vwdm	0.64	0.981	0.887	0.873	0.498	0.627	0.751	0.542	0.602
42	mdd2vwdbow	0.64	0.975	0.884	0.850	0.510	0.626	0.739	0.549	0.596
43	edglw	0.47	0.937	0.865	0.740	0.569	0.621	0.679	0.585	0.562
44	mdw2vcbow	0.65	0.985	0.886	0.886	0.478	0.618	0.753	0.525	0.597
45	cosft	0.82	0.934	0.861	0.686	0.563	0.611	0.652	0.579	0.537
46	edft	0.57	0.917	0.850	0.646	0.576	0.598	0.623	0.582	0.516

#	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
47	mdft	0.71	0.943	0.862	0.709	0.528	0.595	0.655	0.551	0.529
48	mdw2vsg	0.65	0.982	0.880	0.872	0.459	0.594	0.731	0.504	0.574
49	cosgld	0.39	0.887	0.836	0.618	0.627	0.592	0.596	0.607	0.512
50	mdglw	0.63	0.936	0.857	0.737	0.525	0.590	0.663	0.545	0.532
51	edgld	0.23	0.906	0.829	0.601	0.511	0.539	0.569	0.520	0.446
52	mdgld	0.45	0.847	0.791	0.543	0.561	0.518	0.521	0.537	0.410
53	hdp	0.37	0.945	0.836	0.763	0.396	0.481	0.604	0.418	0.452
54	lda	0.71	0.548	0.612	0.321	0.880	0.466	0.367	0.646	0.345
55	cosd2vddm	0.00	0.510	0.533	0.236	0.628	0.343	0.270	0.471	0.109
56	cosd2vddbow	0.00	0.517	0.538	0.237	0.622	0.343	0.271	0.469	0.110
57	mdd2vddm	0.13	0.024	0.209	0.194	0.980	0.324	0.232	0.542	0.007
58	mdd2vddbow	0.13	0.017	0.204	0.193	0.980	0.323	0.230	0.540	-0.004
59	edd2vddbow	0.00	0.484	0.500	0.209	0.563	0.304	0.239	0.420	0.038
60	edd2vddm	0.00	0.473	0.492	0.206	0.568	0.302	0.236	0.420	0.033

Tablica 30. Rezultati prvog ciklusa eksperimenata za korpus MSRP

#	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	wmdft	0.26	0.311	0.732	0.725	0.953	0.823	0.761	0.896	0.364
2	cosbert	0.59	0.321	0.732	0.727	0.949	0.823	0.762	0.894	0.365
3	edbert	0.36	0.327	0.733	0.728	0.947	0.823	0.763	0.893	0.367
4	mdbert	0.53	0.258	0.725	0.713	0.970	0.822	0.753	0.905	0.349
5	tfidf	0.34	0.165	0.706	0.693	0.990	0.815	0.737	0.912	0.303
6	le	0.38	0.207	0.710	0.700	0.974	0.815	0.742	0.904	0.304
7	wmdw2vcbow	0.22	0.224	0.711	0.703	0.968	0.815	0.744	0.900	0.306
8	scsd2vwdbow	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308
9	scsd2vwdm	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308
10	scsft	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308
11	scsglw	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308

#	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
12	scsw2vcbow	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308
13	scsw2vsg	0.48	0.230	0.712	0.705	0.966	0.815	0.745	0.899	0.308
14	cosuse	0.59	0.266	0.716	0.712	0.953	0.815	0.750	0.893	0.318
15	eduse	0.36	0.266	0.716	0.712	0.953	0.815	0.750	0.893	0.318
16	wmdglw	0.19	0.177	0.704	0.694	0.981	0.813	0.737	0.906	0.287
17	cosw2vcbow	0.47	0.240	0.711	0.706	0.959	0.813	0.745	0.895	0.304
18	edd2vwdbow	0.27	0.240	0.711	0.706	0.959	0.813	0.745	0.895	0.304
19	cosd2vwdbow	0.47	0.242	0.711	0.706	0.958	0.813	0.746	0.894	0.303
20	cosd2vwdm	0.47	0.242	0.711	0.706	0.958	0.813	0.746	0.894	0.303
21	mduse	0.56	0.283	0.716	0.714	0.943	0.813	0.751	0.887	0.315
22	edw2vcbow	0.27	0.228	0.708	0.703	0.960	0.812	0.743	0.895	0.292
23	edd2vwdm	0.27	0.230	0.709	0.703	0.960	0.812	0.743	0.895	0.295
24	mdd2vwdbow	0.49	0.262	0.711	0.710	0.948	0.812	0.747	0.888	0.302
25	mdd2vwdm	0.49	0.262	0.711	0.710	0.948	0.812	0.747	0.888	0.302
26	wmdd2vwdbow	0.19	0.171	0.701	0.692	0.980	0.811	0.735	0.905	0.276
27	wmdd2vwdm	0.19	0.171	0.701	0.692	0.980	0.811	0.735	0.905	0.276
28	cosglw	0.43	0.161	0.699	0.690	0.982	0.810	0.734	0.905	0.270
29	wmdw2vsg	0.20	0.179	0.701	0.693	0.975	0.810	0.736	0.902	0.274
30	mdw2vcbow	0.47	0.183	0.701	0.694	0.973	0.810	0.736	0.901	0.273
31	edglw	0.26	0.175	0.698	0.692	0.973	0.809	0.734	0.900	0.263
32	cosgld	0.46	0.224	0.704	0.701	0.956	0.809	0.740	0.891	0.278
33	rp	0.31	0.124	0.690	0.682	0.988	0.807	0.727	0.907	0.244
34	mdft	0.45	0.163	0.695	0.689	0.974	0.807	0.732	0.900	0.250
35	edgld	0.26	0.191	0.697	0.694	0.964	0.807	0.735	0.894	0.257
36	edft	0.25	0.199	0.698	0.695	0.960	0.807	0.736	0.892	0.259
37	cosft	0.44	0.201	0.697	0.695	0.958	0.806	0.736	0.891	0.257
38	mdw2vsg	0.52	0.380	0.717	0.733	0.894	0.806	0.760	0.857	0.326
39	mdglw	0.47	0.150	0.691	0.686	0.975	0.805	0.729	0.900	0.238
40	coselmo	0.73	0.132	0.686	0.682	0.976	0.803	0.725	0.899	0.216

#	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
41	edw2vsg	0.32	0.364	0.712	0.728	0.895	0.803	0.756	0.856	0.311
42	cosw2vsg	0.54	0.366	0.713	0.729	0.895	0.803	0.757	0.856	0.313
43	edelmo	0.47	0.116	0.682	0.678	0.980	0.802	0.723	0.900	0.202
44	jacc	0.21	0.159	0.686	0.685	0.964	0.801	0.727	0.891	0.216
45	mdgld	0.47	0.161	0.687	0.686	0.964	0.801	0.728	0.891	0.219
46	mdelmo	0.59	0.102	0.677	0.675	0.980	0.799	0.720	0.898	0.180
47	lsi	0.39	0.035	0.667	0.663	1.000	0.798	0.711	0.908	0.151
48	coslaser	0.78	0.327	0.702	0.718	0.899	0.798	0.748	0.855	0.280
49	edlaser	0.53	0.325	0.698	0.716	0.894	0.795	0.746	0.852	0.271
50	mdlaser	0.64	0.087	0.669	0.670	0.974	0.794	0.715	0.893	0.139
51	lev	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
52	mdd2vddbow	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
53	mdd2vddm	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
54	gwt	0.08	0.006	0.657	0.657	0.999	0.792	0.705	0.905	0.045
55	lda	0.00	0.004	0.654	0.655	0.996	0.791	0.704	0.902	-0.002
56	hdp	0.05	0.240	0.659	0.688	0.879	0.772	0.719	0.833	0.154
57	cosd2vddbow	0.00	0.502	0.504	0.659	0.505	0.572	0.621	0.530	0.007
58	cosd2vddm	0.00	0.502	0.504	0.659	0.505	0.572	0.621	0.530	0.007
59	edd2vddbow	0.00	0.518	0.508	0.665	0.502	0.572	0.624	0.528	0.019
60	edd2vddm	0.00	0.518	0.508	0.665	0.502	0.572	0.624	0.528	0.019

Tablica 31. Rezultati prvog ciklusa eksperimenata za korpus P4PIN

#	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	cosuse	0.64	0.990	0.970	0.967	0.908	0.937	0.955	0.919	0.918
2	eduse	0.40	0.990	0.970	0.967	0.908	0.937	0.955	0.919	0.918
3	mduse	0.56	0.976	0.967	0.927	0.939	0.933	0.930	0.936	0.911
4	edbert	0.40	0.982	0.963	0.942	0.902	0.922	0.934	0.910	0.898
5	mdbert	0.56	0.972	0.961	0.915	0.926	0.921	0.917	0.924	0.895

#	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
6	cosbert	0.65	0.986	0.963	0.954	0.890	0.921	0.940	0.902	0.897
7	wmdglw	0.13	0.968	0.954	0.902	0.908	0.905	0.904	0.907	0.875
8	wmdd2vwdbow	0.13	0.966	0.952	0.897	0.908	0.902	0.899	0.906	0.871
9	wmdd2vwdm	0.13	0.966	0.952	0.897	0.908	0.902	0.899	0.906	0.871
10	mdlaser	0.70	0.980	0.954	0.934	0.871	0.902	0.921	0.883	0.872
11	wmdft	0.16	0.976	0.952	0.923	0.877	0.899	0.913	0.886	0.869
12	lev	0.33	0.990	0.954	0.965	0.840	0.898	0.937	0.863	0.872
13	wmdw2vcbow	0.14	0.970	0.949	0.906	0.883	0.894	0.901	0.888	0.861
14	jacc	0.17	0.978	0.948	0.927	0.853	0.888	0.911	0.867	0.855
15	edlaser	0.56	0.970	0.939	0.901	0.840	0.870	0.888	0.852	0.831
16	le	0.25	0.978	0.940	0.924	0.822	0.870	0.902	0.841	0.834
17	tfidf	0.25	0.966	0.934	0.889	0.834	0.861	0.877	0.845	0.819
18	coslaser	0.81	0.970	0.934	0.899	0.822	0.859	0.883	0.836	0.818
19	cosgld	0.42	0.968	0.933	0.893	0.822	0.856	0.878	0.835	0.814
20	rp	0.28	0.970	0.930	0.897	0.804	0.848	0.877	0.821	0.805
21	mdgld	0.47	0.984	0.933	0.940	0.773	0.848	0.901	0.802	0.812
22	edgld	0.25	0.986	0.933	0.947	0.767	0.847	0.904	0.797	0.812
23	lsi	0.62	0.939	0.921	0.820	0.865	0.842	0.828	0.856	0.790
24	mdelmo	0.67	0.953	0.916	0.845	0.804	0.824	0.837	0.812	0.770
25	edelmo	0.57	0.935	0.912	0.806	0.840	0.823	0.813	0.833	0.765
26	coselmo	0.82	0.941	0.913	0.818	0.828	0.823	0.820	0.826	0.766
27	gwt	0.07	0.929	0.897	0.783	0.798	0.790	0.786	0.795	0.722
28	scsd2vwdbow	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
29	scsd2vwdm	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
30	scsft	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
31	scsglw	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
32	scsw2vcbow	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
33	scsw2vsg	0.56	0.941	0.896	0.804	0.755	0.778	0.794	0.764	0.711
34	cosglw	0.62	0.951	0.885	0.816	0.681	0.742	0.785	0.704	0.674

#	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
35	wmdw2vsg	0.34	1.000	0.900	1.000	0.589	0.741	0.878	0.642	0.721
36	mdglw	0.56	0.947	0.882	0.804	0.681	0.738	0.776	0.703	0.666
37	edglw	0.38	0.959	0.885	0.836	0.656	0.735	0.793	0.686	0.671
38	edd2vwdbow	0.33	0.905	0.866	0.716	0.742	0.729	0.721	0.737	0.640
39	mdd2vwdbow	0.53	0.907	0.866	0.719	0.736	0.727	0.722	0.733	0.638
40	cosd2vwdbow	0.55	0.901	0.863	0.708	0.742	0.725	0.714	0.735	0.633
41	cosd2vwdm	0.63	0.970	0.884	0.870	0.613	0.719	0.803	0.652	0.664
42	edd2vwdm	0.39	0.970	0.884	0.870	0.613	0.719	0.803	0.652	0.664
43	cosw2vcbow	0.65	0.923	0.866	0.742	0.687	0.713	0.730	0.697	0.627
44	mdd2vwdm	0.57	0.959	0.873	0.825	0.607	0.700	0.770	0.641	0.633
45	lda	0.56	0.677	0.715	0.453	0.834	0.587	0.499	0.714	0.441
46	hdp	0.47	0.744	0.722	0.451	0.656	0.535	0.482	0.602	0.359
47	edw2vcbow	0.52	0.998	0.843	0.983	0.362	0.529	0.732	0.414	0.541
48	mdw2vcbow	0.67	0.996	0.837	0.966	0.344	0.507	0.709	0.394	0.518
49	cosft	0.86	1.000	0.830	1.000	0.301	0.462	0.682	0.350	0.495
50	mdft	0.74	1.000	0.827	1.000	0.288	0.448	0.670	0.336	0.484
51	edft	0.63	1.000	0.825	1.000	0.282	0.440	0.663	0.330	0.479
52	mdd2vddbow	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
53	mdd2vddm	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
54	edd2vddbow	0.00	0.475	0.490	0.246	0.534	0.337	0.276	0.433	0.008
55	edd2vddm	0.00	0.475	0.490	0.246	0.534	0.337	0.276	0.433	0.008
56	cosd2vddbow	0.00	0.469	0.472	0.225	0.479	0.306	0.251	0.390	-0.045
57	cosd2vddm	0.00	0.469	0.472	0.225	0.479	0.306	0.251	0.390	-0.045
58	cosw2vsg	0.94	1.000	0.764	1.000	0.031	0.060	0.137	0.038	0.153
59	edw2vsg	0.93	1.000	0.757	0.000	0.000	0.000	0.000	0.000	0.000
60	mdw2vsg	0.95	1.000	0.757	0.000	0.000	0.000	0.000	0.000	0.000

Tablica 32. Rezultati prvog ciklusa eksperimenata za korpus VMEN

Rank	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
1	le	0.32	0.999	0.998	0.981	0.940	0.958	0.971	0.947	0.958
2	jacc	0.23	1.000	0.998	0.989	0.930	0.957	0.976	0.940	0.957
3	rp	0.38	0.999	0.997	0.981	0.920	0.948	0.967	0.931	0.948
4	lev	0.36	0.999	0.998	0.980	0.920	0.947	0.966	0.931	0.947
5	wmdft	0.41	0.999	0.997	0.962	0.920	0.939	0.953	0.927	0.938
6	wmdd2vwdbow	0.38	0.999	0.997	0.955	0.920	0.935	0.946	0.925	0.934
7	wmdd2vwdm	0.38	0.999	0.997	0.955	0.920	0.935	0.946	0.925	0.934
8	wmdglw	0.38	0.999	0.997	0.955	0.920	0.935	0.946	0.925	0.934
9	wmdw2vcbow	0.38	0.999	0.997	0.955	0.920	0.935	0.946	0.925	0.934
10	cosbert	0.75	1.000	0.997	1.000	0.880	0.935	0.973	0.901	0.936
11	coselmo	0.95	1.000	0.997	1.000	0.880	0.935	0.973	0.901	0.936
12	wmdw2vsg	0.39	0.999	0.997	0.961	0.910	0.934	0.950	0.919	0.933
13	edelmo	0.77	1.000	0.997	0.989	0.880	0.929	0.964	0.898	0.930
14	mdelmo	0.83	1.000	0.997	1.000	0.870	0.929	0.970	0.892	0.930
15	coslaser	0.92	0.999	0.996	0.980	0.880	0.924	0.956	0.897	0.925
16	mdlaser	0.79	0.999	0.996	0.980	0.880	0.924	0.956	0.897	0.925
17	edbert	0.51	1.000	0.996	1.000	0.860	0.923	0.968	0.884	0.925
18	mdbert	0.65	1.000	0.996	0.989	0.860	0.919	0.960	0.882	0.920
19	edlaser	0.73	0.999	0.996	0.979	0.870	0.918	0.953	0.888	0.919
20	lsi	0.40	0.999	0.995	0.970	0.840	0.896	0.938	0.861	0.898
21	tfidf	0.40	0.999	0.995	0.970	0.840	0.896	0.938	0.861	0.898
22	cosuse	0.83	0.999	0.994	0.978	0.780	0.865	0.929	0.811	0.869
23	mduse	0.71	0.999	0.994	0.962	0.790	0.863	0.918	0.817	0.866
24	eduse	0.58	0.999	0.993	0.962	0.780	0.855	0.914	0.807	0.860
25	gwt	0.03	1.000	0.993	0.988	0.730	0.836	0.919	0.768	0.844
26	cosgld	0.55	0.999	0.991	0.958	0.680	0.795	0.885	0.722	0.803
27	scsd2vwdbow	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788
28	scsd2vwdm	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788

Rank	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
29	scsft	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788
30	scsglw	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788
31	scsw2vcbow	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788
32	scsw2vsg	0.87	1.000	0.991	0.982	0.640	0.773	0.885	0.687	0.788
33	edgld	0.33	0.992	0.985	0.780	0.740	0.737	0.757	0.733	0.742
34	edw2vcbow	0.66	0.997	0.987	0.895	0.600	0.713	0.810	0.639	0.724
35	mdd2vwdbow	0.76	0.996	0.986	0.848	0.620	0.702	0.777	0.648	0.711
36	mdgld	0.53	0.988	0.981	0.757	0.710	0.695	0.723	0.691	0.705
37	mdd2vwdm	0.76	0.996	0.985	0.842	0.600	0.684	0.765	0.628	0.696
38	edd2vwdm	0.65	0.991	0.982	0.792	0.640	0.677	0.735	0.646	0.688
39	edd2vwdbow	0.64	0.990	0.981	0.751	0.650	0.670	0.710	0.650	0.676
40	cosd2vwdbow	0.87	0.991	0.982	0.775	0.630	0.670	0.724	0.639	0.678
41	cosd2vwdm	0.88	0.991	0.982	0.792	0.620	0.667	0.730	0.631	0.678
42	mdw2vcbow	0.77	0.998	0.986	0.913	0.520	0.650	0.778	0.564	0.676
43	cosglw	0.89	0.997	0.985	0.882	0.510	0.631	0.754	0.551	0.656
44	edglw	0.68	0.996	0.983	0.828	0.510	0.609	0.714	0.542	0.630
45	cosw2vcbow	0.88	0.991	0.980	0.792	0.550	0.602	0.686	0.561	0.626
46	mdglw	0.77	0.994	0.982	0.827	0.500	0.583	0.690	0.524	0.613
47	cosft	0.95	0.999	0.983	0.892	0.400	0.543	0.703	0.447	0.585
48	edft	0.78	0.999	0.982	0.929	0.350	0.486	0.659	0.393	0.548
49	mdw2vsg	0.81	0.985	0.971	0.840	0.420	0.471	0.618	0.414	0.534
50	mdft	0.84	0.998	0.981	0.920	0.320	0.447	0.616	0.360	0.514
51	cosw2vsg	0.92	0.945	0.932	0.801	0.450	0.408	0.545	0.363	0.482
52	edw2vsg	0.75	0.975	0.958	0.829	0.310	0.286	0.417	0.260	0.384
53	hdp	0.96	0.978	0.957	0.175	0.180	0.168	0.170	0.173	0.151
54	lda	0.92	0.494	0.507	0.049	0.980	0.093	0.060	0.203	0.151
55	edd2vddm	0.10	0.855	0.837	0.038	0.160	0.047	0.040	0.068	0.013
56	edd2vddbow	0.09	0.842	0.825	0.042	0.170	0.046	0.041	0.066	0.013
57	cosd2vddbow	0.18	0.881	0.862	0.028	0.140	0.044	0.032	0.072	0.009

Rank	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
58	mdd2vddbow	0.34	0.721	0.710	0.031	0.290	0.042	0.033	0.069	0.008
59	mdd2vddm	0.34	0.721	0.710	0.031	0.290	0.042	0.033	0.069	0.008
60	cosd2vddm	0.15	0.862	0.843	0.025	0.140	0.042	0.030	0.070	0.001

Tablica 33. Rezultati prvog ciklusa eksperimenata za korpus Webis

Rank	Metoda	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	mdelmo	0.54	0.112	0.560	0.535	0.999	0.697	0.590	0.851	0.241
2	coselmo	0.67	0.113	0.561	0.535	0.999	0.697	0.590	0.851	0.242
3	edelmo	0.41	0.113	0.561	0.535	0.999	0.697	0.590	0.851	0.242
4	lsi	0.11	0.113	0.561	0.535	0.999	0.697	0.590	0.851	0.242
5	mduse	0.41	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
6	cosuse	0.38	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
7	eduse	0.21	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
8	cosbert	0.44	0.129	0.563	0.537	0.989	0.696	0.591	0.846	0.231
9	mdbert	0.46	0.129	0.564	0.537	0.990	0.696	0.591	0.847	0.234
10	edbert	0.25	0.130	0.564	0.537	0.989	0.696	0.591	0.846	0.232
11	edgld	0.12	0.102	0.556	0.532	1.000	0.695	0.587	0.850	0.233
12	wmdd2vwdm	0.00	0.102	0.556	0.532	1.000	0.695	0.587	0.850	0.233
13	cosd2vwdbow	0.13	0.103	0.556	0.533	1.000	0.695	0.587	0.851	0.234
14	edd2vwdbow	0.07	0.103	0.556	0.533	1.000	0.695	0.587	0.851	0.234
15	mdglw	0.41	0.104	0.557	0.533	1.000	0.695	0.588	0.851	0.236
16	jacc	0.04	0.107	0.558	0.533	0.999	0.695	0.588	0.850	0.235
17	tfidf	0.03	0.109	0.558	0.533	0.996	0.695	0.588	0.849	0.230
18	wmdd2vwdbow	0.00	0.111	0.558	0.533	0.995	0.695	0.588	0.848	0.227
19	wmdglw	0.00	0.111	0.558	0.533	0.995	0.695	0.588	0.848	0.227
20	le	0.06	0.115	0.559	0.534	0.994	0.695	0.589	0.848	0.228
21	lev	0.23	0.115	0.559	0.534	0.994	0.695	0.589	0.848	0.228
22	edw2vsg	0.81	0.098	0.554	0.531	1.000	0.694	0.586	0.850	0.228

Rank	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
23	mdw2vsg	0.88	0.098	0.554	0.531	1.000	0.694	0.586	0.850	0.228
24	edglw	0.13	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
25	scsd2vwdbow	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
26	scsd2vwdm	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
27	scsft	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
28	scsglw	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
29	scsw2vcbow	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
30	scsw2vsg	0.10	0.100	0.555	0.532	1.000	0.694	0.587	0.850	0.231
31	cosgld	0.26	0.103	0.555	0.532	0.997	0.694	0.587	0.849	0.226
32	mdgld	0.41	0.106	0.556	0.533	0.997	0.694	0.587	0.849	0.229
33	wmdw2vsg	0.75	0.109	0.557	0.533	0.995	0.694	0.588	0.848	0.226
34	gwt	0.00	0.093	0.551	0.530	1.000	0.693	0.585	0.849	0.222
35	mdd2vwdbow	0.32	0.094	0.552	0.530	1.000	0.693	0.585	0.849	0.223
36	rp	0.03	0.097	0.549	0.529	0.992	0.690	0.583	0.844	0.201
37	coslaser	0.43	0.115	0.554	0.532	0.984	0.690	0.585	0.841	0.199
38	cosft	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
39	edft	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
40	edlaser	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
41	mdd2vddbow	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
42	mdd2vddm	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
43	mdd2vwdm	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
44	mdft	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
45	mdlaser	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
46	wmdft	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
47	cosd2vwdm	0.00	0.077	0.544	0.525	1.000	0.689	0.581	0.847	0.201
48	cosw2vcbow	0.00	0.077	0.544	0.525	1.000	0.689	0.581	0.847	0.201
49	cosw2vsg	0.97	0.077	0.544	0.525	1.000	0.689	0.581	0.847	0.201
50	edw2vcbow	0.88	0.077	0.544	0.525	1.000	0.689	0.581	0.847	0.201
51	mdw2vcbow	0.91	0.077	0.544	0.525	1.000	0.689	0.581	0.847	0.201

Rank	Metoda	θ	S	A	P	R	F1	F.5	F2	MCC
52	cosglw	0.53	0.112	0.552	0.531	0.982	0.689	0.584	0.839	0.192
53	hdp	0.00	0.077	0.543	0.525	0.999	0.688	0.580	0.846	0.197
54	lda	0.03	0.106	0.549	0.529	0.982	0.688	0.583	0.839	0.183
55	wmdw2vcbow	0.61	0.157	0.496	0.501	0.827	0.624	0.544	0.732	-0.021
56	edd2vddm	0.00	0.255	0.465	0.479	0.670	0.559	0.508	0.620	-0.083
57	cosd2vddm	0.00	0.263	0.463	0.478	0.660	0.554	0.506	0.613	-0.084
58	cosd2vddbow	0.00	0.430	0.465	0.472	0.499	0.485	0.477	0.493	-0.072
59	edd2vddbow	0.00	0.398	0.447	0.456	0.495	0.475	0.464	0.487	-0.108
60	edd2vwdm	0.99	0.569	0.291	0.043	0.019	0.026	0.034	0.021	-0.495

Dodatak: Popis 146 modela drugog ciklusa eksperimenata (dokumenti)

U drugom ciklusu druge faze istraživanja spominje se 146 modela temeljenih na metodi dubokog učenja i (uglavnom) na arhitekturi transformera. Nazivi tih 146 modela vidljivi su u cjelovitim rezultatima u poglavlju *Dodatak: Cjeloviti rezultati drugog ciklusa (dokumenti)*, no nije bilo praktično pobrojiti ih u samom tekstu:

M_i={all-MiniLM-L12-v2, distilroberta-base-msmarco-v1, distilroberta-base-msmarco-v2, msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned, msmarco-MiniLM-L-6-v3, msmarco-roberta-base-v3, multi-qa-distilbert-cos-v1, msmarco-distilbert-base-v2, msmarco-distilroberta-base-v2, msmarco-MiniLM-L6-cos-v5, msmarco-roberta-base-v2, multi-qa-mpnet-base-dot-v1, paraphrase-distilroberta-base-v2, all-MiniLM-L12-v1, multi-qa-MiniLM-L6-dot-v1, multi-qa-mpnet-base-cos-v1, jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs, paraphrase-TinyBERT-L6-v2, msmarco-distilbert-base-v4, msmarco-distilbert-cos-v5, paraphrase-mpnet-base-v2, all-distilroberta-v1, msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch, nq-distilbert-base-v1, msmarco-distilbert-dot-v5, jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs, msmarco-distilbert-base-v3, Hoax0930_paraphrase-multilingual-MiniLM-L12-v2, Hoax0930_paraphrase-multilingual-mpnet-base-v2, paraphrase-MiniLM-L12-v2, all-MiniLM-L6-v2, paraphrase-MiniLM-L3-v2, DataikuNLP_paraphrase-MiniLM-L6-v2, msmarco-bert-base-dot-v5, paraphrase-MiniLM-L6-v2, distiluse-base-multilingual-cased-v1, all-roberta-large-v1, DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2, Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2, keithhon_paraphrase-multilingual-MiniLM-L12-v2, paraphrase-multilingual-MiniLM-L12-v2, jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs, paraphrase-albert-base-v2, multi-qa-MiniLM-L6-cos-v1, all-mpnet-base-v1, jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs, jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs, jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs, hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot, jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs, msmarco-distilbert-base-dot-prod-v3, msmarco-bert-co-

condensor, gart-labor_paraphrase-MiniLM-L6-v2-eclass, JoBeer_paraphrase-MiniLM-L6-v2-eclass, all-MiniLM-L6-v1, jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs, paraphrase-multilingual-mpnet-base-v2, distilroberta-base-paraphrase-v1, paraphrase-distilroberta-base-v1, AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2, distilbert-multilingual-nli-stsb-quora-ranking, quora-distilbert-multilingual, hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot, DataikuNLP_paraphrase-albert-small-v2, paraphrase-albert-small-v2, msmarco-MiniLM-L-12-v3, Huffon_paraphrase-multilingual-mpnet-base-v2-512, msmarco-MiniLM-L12-cos-v5, gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2, allenai-specter, multi-qa-distilbert-dot-v1, msmarco-distilbert-base-tas-b, all-mpnet-base-v2, nli-mpnet-base-v2, nli-distilroberta-base-v2, orenpereg_paraphrase-mpnet-base-v2_sst2_64samps, ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8, nli-roberta-base-v2, facebook-dpr-ctx_encoder-single-nq-base, distilbert-base-nli-stsb-mean-tokens, gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2, hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot, JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp, LaBSE, AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2, AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1, facebook-dpr-ctx_encoder-multiset-base, JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass, facebook-dpr-question_encoder-multiset-base, facebook-dpr-question_encoder-single-nq-base, valurank_paraphrase-mpnet-base-v2-offensive, new5558_chula-course-paraphrase-multilingual-mpnet-base-v2, distilbert-base-nli-stsb-wkpooling, average_word_embeddings_komninos, quora-distilbert-base, bert-base-nli-cls-token, nli-bert-base-cls-pooling, roberta-large-nli-stsb-mean-tokens, deutsche-telekom_gbert-large-paraphrase-cosine, deutsche-telekom_gbert-large-paraphrase-euclidean, bert-large-nli-cls-token, nli-bert-large-cls-pooling, bert-base-wikipedia-sections-mean-tokens, average_word_embeddings_glove.6B.300d, distiluse-base-multilingual-cased, distiluse-base-multilingual-cased-v2, hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot, distilbert-base-nli-mean-tokens, bert-base-nli-stsb-mean-tokens, average_word_embeddings_levy_dependency, roberta-base-nli-stsb-mean-tokens, average_word_embeddings_glove.840B.300d, msmarco-roberta-base-ance-firstp, msmarco-roberta-base-ance-fristp, bert-large-nli-stsb-mean-tokens, Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2, nli-distilbert-base-max-pooling, bert-large-nli-mean-tokens, distilbert-base-nli-max-tokens, nli-bert-large, bert-base-nli-mean-tokens, bert-base-nli-

wkpooling, nli-bert-base, distilbert-base-nli-wkpooling, bert-base-nli-max-tokens, nli-bert-base-max-pooling, distilbert-base-nli-stsb-quora-ranking, roberta-large-nli-mean-tokens, bert-large-nli-max-tokens, nli-bert-large-max-pooling, nli-distilbert-base, roberta-base-nli-mean-tokens, nli-roberta-large, nli-roberta-base, Prompsit_paraphrase-bert-en, bert-base-nli-stsb-wkpooling, moshew_paraphrase-mpnet-base-v2_SetFit_emotions, clip-ViT-B-32-multilingual-v1, cointegrated_rubert-base-cased-dp-paraphrase-detection, moshew_paraphrase-mpnet-base-v2_SetFit_sst2, Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2, aditeyabaral-xlm-roberta-base, aditeyabaral-bert-base-cased, aditeyabaral-contrastive-roberta-base, aditeyabaral-distilbert-base-cased, aditeyabaral-roberta-base}

Dodatak: Cjeloviti rezultati drugog ciklusa (dokumenti)

Ovaj dodatak sadrži tablice rezultata za pet korištenih korpusa kao izvora za sažete rezultate predstavljene u tablici 22. Naslovi tablica imaju skraćene naslove: *θ=Threshold* (granična vrijednost); *S=Specificity* (specifičnost); *A=Accuracy* (točnost); *P=Precision* (preciznost); *R=Recall* (Odziv); MCC=*Matthews Correlation Coefficient* (Matthewsov koeficijent korelacije).

Tablica 34. Rezultati jezičnih modela drugog ciklusa eksperimenata za korpus CS

#	Model	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	all-MiniLM-L12-v2	0.47	0.993	0.992	0.970	0.990	0.979	0.973	0.985	0.975
2	distilroberta-base-msmarco-v1	0.39	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
3	distilroberta-base-msmarco-v2	0.48	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
4	msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.40	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
5	msmarco-MiniLM-L-6-v3	0.48	0.993	0.991	0.972	0.982	0.976	0.973	0.979	0.971
6	msmarco-roberta-base-v3	0.34	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
7	multi-qa-distilbert-cos-v1	0.42	0.990	0.991	0.962	0.995	0.976	0.967	0.987	0.972
8	msmarco-distilbert-base-v2	0.32	0.992	0.991	0.965	0.990	0.976	0.969	0.984	0.972
9	msmarco-distilroberta-base-v2	0.48	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
10	msmarco-MiniLM-L6-cos-v5	0.48	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
11	msmarco-roberta-base-v2	0.43	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
12	multi-qa-mpnet-base-dot-v1	0.51	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
13	paraphrase-distilroberta-base-v2	0.43	0.993	0.991	0.970	0.985	0.976	0.972	0.981	0.972
14	all-MiniLM-L12-v1	0.48	0.989	0.990	0.958	0.995	0.974	0.964	0.986	0.970
15	multi-qa-MiniLM-L6-dot-v1	0.68	0.992	0.990	0.965	0.985	0.974	0.968	0.980	0.968
16	multi-qa-mpnet-base-cos-v1	0.41	0.992	0.990	0.965	0.985	0.974	0.968	0.980	0.969
17	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.49	0.990	0.990	0.962	0.990	0.974	0.966	0.983	0.969

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
18	paraphrase-TinyBERT-L6-v2	0.41	0.990	0.990	0.960	0.990	0.974	0.966	0.983	0.969
19	msmarco-distilbert-base-v4	0.40	0.992	0.990	0.966	0.985	0.974	0.969	0.980	0.969
20	msmarco-distilbert-cos-v5	0.40	0.992	0.990	0.966	0.985	0.974	0.969	0.980	0.969
21	paraphrase-mpnet-base-v2	0.46	0.993	0.990	0.969	0.979	0.973	0.971	0.977	0.968
22	all-distilroberta-v1	0.48	0.990	0.989	0.962	0.985	0.971	0.965	0.979	0.966
23	msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.32	0.992	0.989	0.965	0.980	0.971	0.967	0.976	0.965
24	nq-distilbert-base-v1	0.42	0.990	0.989	0.961	0.985	0.971	0.965	0.979	0.966
25	msmarco-distilbert-dot-v5	0.83	0.993	0.989	0.970	0.975	0.971	0.970	0.973	0.965
26	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.55	0.988	0.988	0.953	0.990	0.969	0.959	0.981	0.964
27	msmarco-distilbert-base-v3	0.48	0.990	0.988	0.962	0.980	0.969	0.964	0.975	0.963
28	Hoax0930_paraphrase-multilingual-MiniLM-L12-v2	0.63	0.990	0.988	0.959	0.979	0.968	0.962	0.974	0.962
29	Hoax0930_paraphrase-multilingual-mpnet-base-v2	0.62	0.990	0.988	0.959	0.978	0.968	0.962	0.974	0.961
30	paraphrase-MiniLM-L12-v2	0.48	0.990	0.988	0.959	0.979	0.968	0.962	0.975	0.961
31	all-MiniLM-L6-v2	0.49	0.990	0.988	0.957	0.981	0.968	0.961	0.976	0.962
32	paraphrase-MiniLM-L3-v2	0.47	0.990	0.987	0.960	0.975	0.966	0.962	0.971	0.959
33	DataikuNLP_paraphrase-MiniLM-L6-v2	0.50	0.990	0.987	0.959	0.974	0.966	0.961	0.971	0.958
34	msmarco-bert-base-dot-v5	0.92	0.990	0.987	0.961	0.974	0.966	0.962	0.970	0.959
35	paraphrase-MiniLM-L6-v2	0.50	0.990	0.987	0.959	0.974	0.966	0.961	0.971	0.958
36	distiluse-base-multilingual-cased-v1	0.45	0.990	0.987	0.962	0.975	0.966	0.963	0.971	0.960
37	all-roberta-large-v1	0.50	0.992	0.987	0.965	0.970	0.965	0.964	0.967	0.959
38	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.52	0.992	0.987	0.964	0.969	0.965	0.964	0.967	0.958
39	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.53	0.992	0.987	0.964	0.969	0.965	0.964	0.967	0.958
40	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.52	0.992	0.987	0.964	0.969	0.965	0.964	0.967	0.958
41	paraphrase-multilingual-MiniLM-L12-v2	0.52	0.992	0.987	0.964	0.969	0.965	0.964	0.967	0.958
42	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs	0.55	0.992	0.987	0.965	0.969	0.965	0.965	0.967	0.959
43	paraphrase-albert-base-v2	0.51	0.993	0.987	0.970	0.965	0.965	0.967	0.964	0.959
44	multi-qa-MiniLM-L6-cos-v1	0.40	0.987	0.986	0.951	0.985	0.964	0.956	0.976	0.959
45	all-mpnet-base-v1	0.50	0.990	0.986	0.962	0.970	0.963	0.962	0.966	0.957

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
46	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.54	0.989	0.986	0.956	0.974	0.963	0.958	0.969	0.956
47	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.52	0.992	0.986	0.963	0.964	0.962	0.962	0.963	0.955
48	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.52	0.992	0.986	0.963	0.963	0.962	0.962	0.963	0.954
49	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.52	0.993	0.986	0.968	0.958	0.962	0.965	0.959	0.954
50	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs	0.54	0.992	0.986	0.965	0.961	0.962	0.964	0.961	0.955
51	msmarco-distilbert-base-dot-prod-v3	0.71	0.987	0.985	0.947	0.980	0.961	0.952	0.972	0.954
52	msmarco-bert-co-condensor	0.90	0.989	0.985	0.956	0.970	0.961	0.958	0.966	0.953
53	gart-labor_paraphrase-MiniLM-L6-v2-eclass	0.54	0.992	0.985	0.965	0.960	0.960	0.962	0.959	0.953
54	JoBeer_paraphrase-MiniLM-L6-v2-eclass	0.54	0.992	0.985	0.965	0.960	0.960	0.962	0.959	0.953
55	all-MiniLM-L6-v1	0.49	0.987	0.984	0.950	0.975	0.959	0.952	0.967	0.952
56	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.48	0.987	0.984	0.948	0.974	0.959	0.952	0.967	0.951
57	paraphrase-multilingual-mpnet-base-v2	0.48	0.987	0.984	0.948	0.974	0.959	0.952	0.967	0.951
58	distilroberta-base-paraphrase-v1	0.49	0.987	0.984	0.948	0.975	0.959	0.952	0.968	0.951
59	paraphrase-distilroberta-base-v1	0.49	0.987	0.984	0.948	0.975	0.959	0.952	0.968	0.951
60	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.43	0.990	0.984	0.960	0.959	0.957	0.959	0.958	0.949
61	distilbert-multilingual-nli-stsb-quora-ranking	0.90	0.990	0.984	0.957	0.960	0.957	0.957	0.959	0.949
62	quora-distilbert-multilingual	0.90	0.990	0.984	0.957	0.960	0.957	0.957	0.959	0.949
63	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot	0.55	0.992	0.984	0.963	0.949	0.956	0.960	0.951	0.946
64	DataikuNLP_paraphrase-albert-small-v2	0.45	0.990	0.983	0.962	0.955	0.955	0.958	0.954	0.947
65	paraphrase-albert-small-v2	0.45	0.990	0.983	0.962	0.955	0.955	0.958	0.954	0.947
66	msmarco-MiniLM-L-12-v3	0.48	0.985	0.982	0.944	0.970	0.954	0.947	0.963	0.945
67	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.50	0.986	0.981	0.942	0.963	0.951	0.945	0.958	0.941
68	msmarco-MiniLM-L12-cos-v5	0.48	0.983	0.980	0.937	0.970	0.949	0.941	0.960	0.940
69	gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.48	0.987	0.980	0.944	0.951	0.946	0.944	0.949	0.935
70	allenai-specter	0.88	0.986	0.979	0.946	0.955	0.946	0.945	0.950	0.936
71	multi-qa-distilbert-dot-v1	0.52	0.989	0.980	0.952	0.944	0.946	0.950	0.945	0.936
72	msmarco-distilbert-base-tas-b	0.84	0.984	0.979	0.939	0.960	0.946	0.941	0.954	0.936
73	all-mpnet-base-v2	0.58	0.986	0.978	0.943	0.945	0.942	0.942	0.943	0.930
74	nli-mpnet-base-v2	0.66	0.982	0.977	0.934	0.960	0.942	0.936	0.951	0.931

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
75	nli-distilroberta-base-v2	0.70	0.983	0.977	0.942	0.955	0.942	0.940	0.948	0.933
76	orenpereg_paraphrase-mpnet-base-v2_sst2_64samps	0.77	0.987	0.977	0.950	0.935	0.939	0.945	0.936	0.928
77	ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8	0.70	0.979	0.974	0.916	0.951	0.932	0.922	0.943	0.917
78	nli-roberta-base-v2	0.73	0.981	0.972	0.934	0.940	0.929	0.930	0.933	0.918
79	facebook-dpr-ctx_encoder-single-nq-base	0.82	0.984	0.972	0.934	0.921	0.925	0.930	0.922	0.910
80	distilbert-base-nli-stsb-mean-tokens	0.61	0.990	0.972	0.954	0.895	0.923	0.941	0.906	0.907
81	gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.72	0.989	0.971	0.949	0.899	0.921	0.937	0.907	0.905
82	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot	0.64	0.984	0.968	0.926	0.902	0.913	0.921	0.906	0.894
83	JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp	0.61	0.977	0.964	0.911	0.909	0.905	0.908	0.906	0.887
84	LaBSE	0.70	0.983	0.964	0.935	0.890	0.905	0.921	0.894	0.888
85	AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2	0.59	0.982	0.962	0.922	0.879	0.898	0.912	0.887	0.877
86	AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1	0.56	0.976	0.959	0.903	0.894	0.893	0.898	0.892	0.872
87	facebook-dpr-ctx_encoder-multiset-base	0.80	0.975	0.959	0.896	0.890	0.891	0.894	0.890	0.867
88	JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass	0.47	0.976	0.958	0.905	0.887	0.891	0.898	0.888	0.869
89	facebook-dpr-question_encoder-multiset-base	0.84	0.967	0.955	0.894	0.910	0.889	0.890	0.897	0.872
90	facebook-dpr-question_encoder-single-nq-base	0.88	0.966	0.954	0.883	0.910	0.886	0.882	0.897	0.866
91	valurank_paraphrase-mpnet-base-v2-offensive	0.95	0.963	0.951	0.870	0.904	0.880	0.873	0.892	0.855
92	new5558_chula-course-paraphrase-multilingual-mpnet-base-v2	0.53	0.971	0.953	0.889	0.881	0.880	0.884	0.879	0.855
93	distilbert-base-nli-stsb-wkpooling	0.67	0.979	0.953	0.917	0.848	0.874	0.897	0.856	0.851
94	average_word_embeddings_komninos	0.91	0.968	0.947	0.886	0.865	0.865	0.875	0.862	0.841
95	quora-distilbert-base	0.60	0.966	0.947	0.866	0.874	0.865	0.865	0.870	0.836
96	bert-base-nli-cls-token	0.79	0.965	0.946	0.871	0.875	0.864	0.867	0.868	0.838
97	nli-bert-base-cls-pooling	0.79	0.965	0.946	0.871	0.875	0.864	0.867	0.868	0.838
98	roberta-large-nli-stsb-mean-tokens	0.56	0.957	0.941	0.839	0.875	0.853	0.844	0.865	0.819
99	deutsche-telekom_gbert-large-paraphrase-cosine	0.72	0.975	0.945	0.897	0.825	0.852	0.876	0.834	0.825
100	deutsche-telekom_gbert-large-paraphrase-euclidean	0.77	0.971	0.943	0.881	0.830	0.848	0.865	0.836	0.819
101	bert-large-nli-cls-token	0.79	0.966	0.940	0.878	0.839	0.847	0.863	0.839	0.819
102	nli-bert-large-cls-pooling	0.79	0.966	0.940	0.878	0.839	0.847	0.863	0.839	0.819
103	bert-base-wikipedia-sections-mean-tokens	0.99	0.989	0.946	0.943	0.767	0.844	0.900	0.796	0.820

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
104	average_word_embeddings_glove.6B.300d	0.87	0.975	0.942	0.895	0.807	0.842	0.871	0.819	0.813
105	distiluse-base-multilingual-cased	0.68	0.977	0.942	0.894	0.800	0.841	0.871	0.815	0.810
106	distiluse-base-multilingual-cased-v2	0.68	0.977	0.942	0.894	0.800	0.841	0.871	0.815	0.810
107	hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot	0.80	0.978	0.943	0.893	0.797	0.840	0.870	0.813	0.809
108	distilbert-base-nli-mean-tokens	0.72	0.971	0.937	0.867	0.798	0.830	0.852	0.810	0.794
109	bert-base-nli-stsb-mean-tokens	0.69	0.975	0.938	0.884	0.782	0.829	0.860	0.800	0.794
110	average_word_embeddings_levy_dependency	0.94	0.953	0.929	0.848	0.838	0.827	0.837	0.828	0.796
111	roberta-base-nli-stsb-mean-tokens	0.58	0.939	0.926	0.806	0.874	0.825	0.811	0.851	0.791
112	average_word_embeddings_glove.840B.300d	0.89	0.964	0.933	0.870	0.809	0.825	0.849	0.812	0.795
113	msmarco-roberta-base-ance-firstp	0.99	0.893	0.910	0.722	0.985	0.822	0.757	0.907	0.792
114	msmarco-roberta-base-ance-fristp	0.99	0.893	0.910	0.722	0.985	0.822	0.757	0.907	0.792
115	bert-large-nli-stsb-mean-tokens	0.57	0.956	0.930	0.841	0.826	0.822	0.830	0.821	0.788
116	Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2	0.49	0.984	0.934	0.912	0.727	0.808	0.867	0.757	0.777
117	nli-distilbert-base-max-pooling	0.71	0.958	0.918	0.834	0.753	0.782	0.809	0.762	0.740
118	bert-large-nli-mean-tokens	0.77	0.954	0.915	0.816	0.753	0.773	0.795	0.759	0.730
119	distilbert-base-nli-max-tokens	0.85	0.949	0.913	0.812	0.768	0.773	0.792	0.766	0.732
120	nli-bert-large	0.77	0.954	0.915	0.816	0.753	0.773	0.795	0.759	0.730
121	bert-base-nli-mean-tokens	0.77	0.959	0.914	0.811	0.729	0.767	0.793	0.743	0.717
122	bert-base-nli-wkpooling	0.77	0.959	0.914	0.811	0.729	0.767	0.793	0.743	0.717
123	nli-bert-base	0.77	0.959	0.914	0.811	0.729	0.767	0.793	0.743	0.717
124	distilbert-base-nli-wkpooling	0.77	0.933	0.905	0.764	0.791	0.765	0.761	0.777	0.716
125	bert-base-nli-max-tokens	0.78	0.949	0.910	0.781	0.754	0.763	0.773	0.756	0.711
126	nli-bert-base-max-pooling	0.78	0.949	0.910	0.781	0.754	0.763	0.773	0.756	0.711
127	distilbert-base-nli-stsb-quora-ranking	0.67	0.941	0.902	0.816	0.750	0.754	0.785	0.743	0.717
128	roberta-large-nli-mean-tokens	0.72	0.929	0.900	0.742	0.780	0.754	0.745	0.767	0.697
129	bert-large-nli-max-tokens	0.80	0.934	0.901	0.747	0.770	0.749	0.746	0.759	0.695
130	nli-bert-large-max-pooling	0.80	0.934	0.901	0.747	0.770	0.749	0.746	0.759	0.695
131	nli-distilbert-base	0.78	0.936	0.896	0.774	0.735	0.739	0.756	0.732	0.687
132	roberta-base-nli-mean-tokens	0.78	0.953	0.893	0.795	0.649	0.701	0.750	0.666	0.651

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
133	nli-roberta-large	0.76	0.935	0.883	0.728	0.669	0.690	0.710	0.676	0.624
134	nli-roberta-base	0.82	0.949	0.883	0.835	0.618	0.683	0.761	0.635	0.645
135	Prompsit_paraphrase-bert-en	0.95	0.926	0.863	0.774	0.611	0.653	0.715	0.618	0.600
136	bert-base-nli-stsb-wkpooling	0.74	0.921	0.863	0.686	0.627	0.639	0.661	0.629	0.567
137	moshew_paraphrase-mpnet-base-v2_SetFit_emotions	0.85	0.888	0.835	0.613	0.608	0.601	0.606	0.602	0.505
138	clip-ViT-B-32-multilingual-v1	0.99	0.857	0.820	0.545	0.666	0.594	0.562	0.633	0.489
139	cointegrated_rubert-base-cased-dp-paraphrase-detection	0.95	0.887	0.820	0.567	0.546	0.542	0.553	0.541	0.442
140	moshew_paraphrase-mpnet-base-v2_SetFit_sst2	0.97	0.819	0.764	0.534	0.542	0.481	0.497	0.501	0.378
141	Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2	0.62	0.706	0.679	0.393	0.569	0.422	0.397	0.483	0.265
142	aditeyabaral-xlm-roberta-base	1.00	0.177	0.314	0.205	0.881	0.332	0.242	0.530	0.065
143	aditeyabaral-bert-base-cased	0.00	0.000	0.194	0.194	1.000	0.325	0.231	0.546	0.000
144	aditeyabaral-contrastive-roberta-base	0.00	0.000	0.194	0.194	1.000	0.325	0.231	0.546	0.000
145	aditeyabaral-distilbert-base-cased	0.00	0.000	0.194	0.194	1.000	0.325	0.231	0.546	0.000
146	aditeyabaral-roberta-base	0.00	0.000	0.194	0.194	1.000	0.325	0.231	0.546	0.000

Tablica 35. Rezultati jezičnih modela drugog ciklusa eksperimenata za korpus MSRP

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
1	roberta-base-nli-stsb-mean-tokens	0.59	0.407	0.768	0.754	0.957	0.844	0.788	0.908	0.462
2	bert-base-nli-stsb-mean-tokens	0.60	0.423	0.767	0.757	0.948	0.842	0.789	0.902	0.458
3	AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1	0.59	0.425	0.768	0.758	0.948	0.842	0.790	0.903	0.460
4	roberta-large-nli-stsb-mean-tokens	0.61	0.447	0.767	0.763	0.935	0.840	0.792	0.895	0.457
5	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.70	0.376	0.757	0.745	0.957	0.838	0.779	0.906	0.435
6	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.62	0.435	0.763	0.759	0.935	0.838	0.789	0.893	0.446
7	distilbert-base-nli-stsb-mean-tokens	0.57	0.364	0.753	0.741	0.957	0.835	0.776	0.905	0.423
8	bert-large-nli-stsb-mean-tokens	0.56	0.368	0.753	0.742	0.955	0.835	0.777	0.903	0.423
9	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.70	0.378	0.754	0.744	0.952	0.835	0.778	0.902	0.426
10	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.73	0.437	0.758	0.758	0.926	0.834	0.786	0.887	0.432
11	paraphrase-mpnet-base-v2	0.69	0.417	0.755	0.753	0.933	0.833	0.783	0.890	0.425

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
12	AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2	0.54	0.315	0.744	0.729	0.969	0.832	0.767	0.909	0.403
13	distilbert-base-nli-stsb-wkpooling	0.57	0.350	0.746	0.736	0.955	0.832	0.772	0.902	0.406
14	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.70	0.394	0.751	0.747	0.939	0.832	0.779	0.893	0.416
15	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.70	0.394	0.751	0.747	0.939	0.832	0.779	0.893	0.416
16	paraphrase-multilingual-mpnet-base-v2	0.70	0.394	0.751	0.747	0.939	0.832	0.779	0.893	0.416
17	nli-roberta-base-v2	0.74	0.437	0.755	0.757	0.922	0.831	0.785	0.883	0.425
18	paraphrase-MiniLM-L12-v2	0.60	0.321	0.739	0.729	0.958	0.828	0.765	0.902	0.385
19	bert-base-nli-stsb-wkpooling	0.57	0.388	0.746	0.744	0.934	0.828	0.775	0.888	0.401
20	paraphrase-distilroberta-base-v2	0.57	0.293	0.734	0.722	0.966	0.826	0.760	0.905	0.374
21	all-mpnet-base-v1	0.65	0.325	0.737	0.729	0.954	0.826	0.765	0.899	0.380
22	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.63	0.329	0.737	0.730	0.951	0.826	0.765	0.896	0.377
23	all-mpnet-base-v2	0.68	0.437	0.749	0.755	0.912	0.826	0.782	0.876	0.409
24	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.63	0.325	0.736	0.729	0.952	0.825	0.764	0.897	0.376
25	all-roberta-large-v1	0.67	0.364	0.739	0.737	0.937	0.825	0.770	0.889	0.384
26	Hoax0930_paraphrase-multilingual-mpnet-base-v2	0.62	0.283	0.730	0.719	0.966	0.824	0.758	0.904	0.363
27	nli-mpnet-base-v2	0.70	0.358	0.737	0.735	0.937	0.824	0.768	0.888	0.378
28	bert-large-nli-max-tokens	0.82	0.380	0.739	0.740	0.928	0.824	0.771	0.883	0.383
29	nli-bert-large-max-pooling	0.82	0.380	0.739	0.740	0.928	0.824	0.771	0.883	0.383
30	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.61	0.283	0.728	0.719	0.963	0.823	0.757	0.901	0.356
31	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.61	0.283	0.728	0.719	0.963	0.823	0.757	0.901	0.356
32	paraphrase-multilingual-MiniLM-L12-v2	0.61	0.283	0.728	0.719	0.963	0.823	0.757	0.901	0.356
33	DataikuNLP_paraphrase-albert-small-v2	0.57	0.317	0.732	0.726	0.950	0.823	0.762	0.895	0.363
34	paraphrase-albert-small-v2	0.57	0.317	0.732	0.726	0.950	0.823	0.762	0.895	0.363
35	distilroberta-base-paraphrase-v1	0.59	0.321	0.732	0.727	0.949	0.823	0.762	0.894	0.365
36	paraphrase-distilroberta-base-v1	0.59	0.321	0.732	0.727	0.949	0.823	0.762	0.894	0.365
37	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot	0.65	0.337	0.734	0.730	0.942	0.823	0.765	0.891	0.368
38	all-distilroberta-v1	0.64	0.350	0.736	0.733	0.939	0.823	0.767	0.889	0.374
39	new5558_chula-course-paraphrase-multilingual-mpnet-base-v2	0.66	0.348	0.734	0.732	0.937	0.822	0.766	0.887	0.368

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
40	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.60	0.260	0.723	0.713	0.967	0.821	0.753	0.903	0.343
41	roberta-large-nli-mean-tokens	0.75	0.274	0.725	0.716	0.962	0.821	0.755	0.900	0.345
42	DataikuNLP_paraphrase-MiniLM-L6-v2	0.60	0.301	0.728	0.721	0.952	0.821	0.758	0.895	0.351
43	paraphrase-MiniLM-L6-v2	0.60	0.301	0.728	0.721	0.952	0.821	0.758	0.895	0.351
44	bert-large-nli-mean-tokens	0.79	0.333	0.732	0.729	0.941	0.821	0.763	0.889	0.362
45	nli-bert-large	0.79	0.333	0.732	0.729	0.941	0.821	0.763	0.889	0.362
46	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs	0.61	0.256	0.722	0.712	0.967	0.820	0.752	0.902	0.339
47	nli-roberta-large	0.75	0.260	0.722	0.713	0.965	0.820	0.752	0.901	0.338
48	nli-roberta-base	0.77	0.299	0.727	0.721	0.952	0.820	0.758	0.895	0.349
49	paraphrase-TinyBERT-L6-v2	0.59	0.346	0.732	0.731	0.935	0.820	0.764	0.885	0.362
50	nli-distilroberta-base-v2	0.69	0.309	0.726	0.722	0.946	0.819	0.758	0.891	0.346
51	roberta-base-nli-mean-tokens	0.78	0.315	0.726	0.724	0.942	0.819	0.759	0.889	0.346
52	all-MiniLM-L12-v2	0.64	0.323	0.728	0.725	0.940	0.819	0.760	0.888	0.350
53	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs	0.61	0.240	0.718	0.708	0.969	0.818	0.748	0.902	0.327
54	distilbert-base-nli-stsb-quora-ranking	0.62	0.307	0.724	0.721	0.943	0.818	0.757	0.889	0.340
55	quora-distilbert-base	0.62	0.307	0.724	0.721	0.943	0.818	0.757	0.889	0.340
56	LaBSE	0.64	0.319	0.725	0.724	0.939	0.818	0.759	0.886	0.344
57	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot	0.55	0.236	0.716	0.707	0.969	0.817	0.747	0.902	0.322
58	Hoax0930_paraphrase-multilingual-MiniLM-L12-v2	0.55	0.240	0.716	0.708	0.967	0.817	0.748	0.901	0.321
59	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.57	0.242	0.716	0.708	0.966	0.817	0.748	0.900	0.321
60	Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2	0.32	0.264	0.719	0.712	0.958	0.817	0.751	0.896	0.327
61	JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass	0.66	0.325	0.725	0.725	0.935	0.817	0.759	0.884	0.342
62	bert-base-nli-max-tokens	0.77	0.329	0.726	0.726	0.935	0.817	0.760	0.884	0.346
63	nli-bert-base-max-pooling	0.77	0.329	0.726	0.726	0.935	0.817	0.760	0.884	0.346
64	bert-large-nli-cls-token	0.83	0.366	0.729	0.734	0.920	0.817	0.765	0.876	0.355
65	nli-bert-large-cls-pooling	0.83	0.366	0.729	0.734	0.920	0.817	0.765	0.876	0.355
66	bert-base-nli-cls-token	0.80	0.297	0.721	0.719	0.944	0.816	0.755	0.889	0.332

#	Model	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
67	nli-bert-base-cls-pooling	0.80	0.297	0.721	0.719	0.944	0.816	0.755	0.889	0.332
68	bert-base-nli-mean-tokens	0.77	0.360	0.728	0.733	0.922	0.816	0.764	0.877	0.352
69	bert-base-nli-wkpooling	0.77	0.360	0.728	0.733	0.922	0.816	0.764	0.877	0.352
70	nli-bert-base	0.77	0.360	0.728	0.733	0.922	0.816	0.764	0.877	0.352
71	paraphrase-albert-base-v2	0.56	0.222	0.712	0.703	0.970	0.815	0.744	0.902	0.309
72	distiluse-base-multilingual-cased-v1	0.57	0.228	0.712	0.704	0.967	0.815	0.745	0.900	0.308
73	distilbert-base-nli-mean-tokens	0.75	0.280	0.717	0.715	0.947	0.814	0.751	0.889	0.319
74	nli-distilbert-base	0.75	0.280	0.717	0.715	0.947	0.814	0.751	0.889	0.319
75	distiluse-base-multilingual-cased-v2	0.66	0.366	0.726	0.733	0.916	0.814	0.764	0.872	0.347
76	distiluse-base-multilingual-cased	0.66	0.366	0.726	0.733	0.916	0.814	0.764	0.872	0.347
77	all-MiniLM-L12-v1	0.65	0.372	0.727	0.735	0.913	0.814	0.764	0.871	0.349
78	distilbert-base-nli-wkpooling	0.73	0.217	0.708	0.701	0.966	0.813	0.742	0.898	0.294
79	distilbert-base-nli-max-tokens	0.82	0.252	0.712	0.708	0.954	0.813	0.747	0.892	0.305
80	paraphrase-MiniLM-L3-v2	0.60	0.274	0.714	0.713	0.946	0.813	0.749	0.887	0.310
81	JoBeer_paraphrase-MiniLM-L6-v2-eclass	0.67	0.287	0.715	0.715	0.940	0.812	0.751	0.884	0.312
82	gart-labor_paraphrase-MiniLM-L6-v2-eclass	0.67	0.287	0.715	0.715	0.940	0.812	0.751	0.884	0.312
83	nli-distilbert-base-max-pooling	0.73	0.333	0.720	0.725	0.923	0.812	0.757	0.875	0.328
84	gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.48	0.177	0.701	0.693	0.976	0.811	0.736	0.903	0.274
85	gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.57	0.240	0.708	0.705	0.954	0.811	0.744	0.891	0.291
86	all-MiniLM-L6-v1	0.61	0.268	0.711	0.711	0.944	0.811	0.748	0.886	0.302
87	orenpereg_paraphrase-mpnet-base-v2_sst2_64samps	0.72	0.274	0.712	0.712	0.942	0.811	0.748	0.885	0.304
88	hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot	0.52	0.175	0.700	0.692	0.975	0.810	0.735	0.902	0.269
89	distilbert-multilingual-nli-stsb-quora-ranking	0.90	0.232	0.706	0.703	0.955	0.810	0.742	0.891	0.285
90	quora-distilbert-multilingual	0.90	0.232	0.706	0.703	0.955	0.810	0.742	0.891	0.285
91	msmarco-distilbert-base-v4	0.52	0.175	0.698	0.692	0.973	0.809	0.734	0.900	0.263
92	msmarco-distilbert-cos-v5	0.52	0.175	0.698	0.692	0.973	0.809	0.734	0.900	0.263
93	all-MiniLM-L6-v2	0.56	0.187	0.700	0.694	0.969	0.809	0.736	0.898	0.266
94	multi-qa-MiniLM-L6-dot-v1	0.75	0.189	0.700	0.694	0.968	0.809	0.736	0.897	0.266
95	msmarco-distilbert-base-v3	0.54	0.224	0.704	0.701	0.956	0.809	0.740	0.891	0.278

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
96	nq-distilbert-base-v1	0.55	0.224	0.703	0.701	0.955	0.808	0.740	0.890	0.275
97	msmarco-bert-co-condensor	0.92	0.165	0.695	0.689	0.973	0.807	0.732	0.899	0.250
98	facebook-dpr-question_encoder-multiset-base	0.85	0.197	0.698	0.695	0.962	0.807	0.736	0.893	0.259
99	multi-qa-distilbert-dot-v1	0.59	0.177	0.695	0.691	0.968	0.806	0.733	0.896	0.251
100	Prompsit_paraphrase-bert-en	0.88	0.183	0.696	0.692	0.966	0.806	0.734	0.895	0.253
101	deutsche-telekom_gbert-large-paraphrase-cosine	0.65	0.270	0.705	0.709	0.934	0.806	0.745	0.878	0.282
102	multi-qa-MiniLM-L6-cos-v1	0.55	0.159	0.691	0.687	0.971	0.805	0.730	0.897	0.236
103	msmarco-distilbert-base-v2	0.59	0.256	0.702	0.706	0.937	0.805	0.742	0.879	0.273
104	multi-qa-distilbert-cos-v1	0.56	0.138	0.688	0.683	0.976	0.804	0.727	0.899	0.224
105	msmarco-distilbert-base-tas-b	0.87	0.207	0.695	0.696	0.952	0.804	0.735	0.887	0.249
106	msmarco-distilbert-base-dot-prod-v3	0.72	0.238	0.699	0.701	0.941	0.804	0.739	0.881	0.261
107	facebook-dpr-question_encoder-single-nq-base	0.89	0.276	0.703	0.709	0.927	0.804	0.744	0.874	0.277
108	moshew_paraphrase-mpnet-base-v2_SetFit_emotions	0.34	0.114	0.684	0.679	0.984	0.803	0.724	0.903	0.214
109	msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.57	0.136	0.686	0.682	0.975	0.803	0.726	0.898	0.218
110	msmarco-bert-base-dot-v5	0.92	0.142	0.687	0.683	0.973	0.803	0.727	0.897	0.220
111	distilroberta-base-msmarco-v1	0.48	0.100	0.681	0.676	0.986	0.802	0.721	0.903	0.200
112	msmarco-roberta-base-v3	0.43	0.128	0.683	0.680	0.975	0.802	0.724	0.898	0.207
113	distilroberta-base-msmarco-v2	0.47	0.132	0.685	0.681	0.975	0.802	0.725	0.898	0.213
114	msmarco-distilroberta-base-v2	0.47	0.132	0.685	0.681	0.975	0.802	0.725	0.898	0.213
115	multi-qa-mpnet-base-dot-v1	0.67	0.173	0.690	0.689	0.962	0.802	0.730	0.891	0.229
116	ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8	0.35	0.059	0.674	0.669	0.998	0.801	0.716	0.908	0.185
117	cointegrated_rubert-base-cased-dp-paraphrase-detection	0.86	0.112	0.681	0.677	0.980	0.801	0.722	0.899	0.196
118	deutsche-telekom_gbert-large-paraphrase-euclidean	0.71	0.291	0.702	0.711	0.918	0.801	0.745	0.867	0.274
119	moshew_paraphrase-mpnet-base-v2_SetFit_sst2	0.04	0.122	0.681	0.679	0.974	0.800	0.722	0.896	0.195
120	msmarco-distilbert-dot-v5	0.85	0.195	0.689	0.692	0.949	0.800	0.731	0.883	0.227
121	msmarco-roberta-base-ance-firstp	0.99	0.126	0.680	0.679	0.971	0.799	0.722	0.894	0.191
122	msmarco-roberta-base-ance-fristp	0.99	0.126	0.680	0.679	0.971	0.799	0.722	0.894	0.191
123	multi-qa-mpnet-base-cos-v1	0.59	0.134	0.681	0.680	0.968	0.799	0.723	0.892	0.194
124	msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.50	0.150	0.682	0.683	0.962	0.799	0.725	0.889	0.200

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
125	bert-base-wikipedia-sections-mean-tokens	0.99	0.061	0.671	0.668	0.991	0.798	0.714	0.904	0.155
126	msmarco-roberta-base-v2	0.49	0.104	0.676	0.675	0.978	0.798	0.719	0.897	0.176
127	JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp	0.66	0.254	0.693	0.702	0.924	0.798	0.738	0.869	0.246
128	msmarco-MiniLM-L-6-v3	0.56	0.224	0.688	0.696	0.933	0.797	0.733	0.873	0.228
129	msmarco-MiniLM-L6-cos-v5	0.56	0.224	0.688	0.696	0.933	0.797	0.733	0.873	0.228
130	average_word_embeddings_glove.840B.300d	0.66	0.112	0.674	0.675	0.970	0.796	0.719	0.892	0.166
131	facebook-dpr-ctx_encoder-single-nq-base	0.66	0.030	0.664	0.662	0.997	0.795	0.709	0.905	0.116
132	facebook-dpr-ctx_encoder-multiset-base	0.73	0.043	0.665	0.663	0.991	0.795	0.710	0.902	0.115
133	Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2	0.26	0.185	0.682	0.688	0.943	0.795	0.727	0.878	0.203
134	msmarco-MiniLM-L-12-v3	0.53	0.126	0.673	0.676	0.960	0.794	0.719	0.886	0.162
135	msmarco-MiniLM-L12-cos-v5	0.53	0.126	0.673	0.676	0.960	0.794	0.719	0.886	0.162
136	aditeyabaral-bert-base-cased	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
137	aditeyabaral-contrastive-roberta-base	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
138	aditeyabaral-distilbert-base-cased	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
139	aditeyabaral-roberta-base	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
140	aditeyabaral-xlm-roberta-base	0.00	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
141	clip-ViT-B-32-multilingual-v1	0.71	0.000	0.655	0.655	1.000	0.792	0.704	0.905	0.000
142	valurank_paraphrase-mpnet-base-v2-offensive	0.44	0.006	0.656	0.656	0.998	0.792	0.705	0.904	0.032
143	allenai-specter	0.74	0.026	0.659	0.660	0.991	0.792	0.707	0.901	0.071
144	average_word_embeddings_komninos	0.72	0.002	0.655	0.655	0.998	0.791	0.704	0.903	-0.001
145	average_word_embeddings_levy_dependency	0.80	0.033	0.658	0.660	0.986	0.791	0.707	0.897	0.063
146	average_word_embeddings_glove.6B.300d	0.67	0.022	0.655	0.658	0.987	0.789	0.705	0.897	0.036

Tablica 36. Rezultati jezičnih modela drugog ciklusa drugog ciklusa eksperimenata za korpus P4PIN

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
1	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.61	0.996	0.990	0.988	0.969	0.978	0.984	0.973	0.972
2	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.75	0.992	0.984	0.975	0.957	0.966	0.971	0.961	0.955
3	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.75	0.992	0.984	0.975	0.957	0.966	0.971	0.961	0.955

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
4	paraphrase-multilingual-mpnet-base-v2	0.75	0.992	0.984	0.975	0.957	0.966	0.971	0.961	0.955
5	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.75	0.986	0.982	0.958	0.969	0.963	0.960	0.967	0.952
6	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.74	0.990	0.981	0.969	0.951	0.960	0.965	0.954	0.947
7	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.74	0.984	0.979	0.952	0.963	0.957	0.954	0.961	0.944
8	distiluse-base-multilingual-cased-v1	0.60	0.988	0.979	0.963	0.951	0.957	0.960	0.953	0.943
9	roberta-large-nli-stsb-mean-tokens	0.69	0.990	0.978	0.968	0.939	0.953	0.962	0.944	0.939
10	LaBSE	0.64	0.982	0.976	0.945	0.957	0.951	0.948	0.955	0.935
11	paraphrase-mpnet-base-v2	0.72	0.984	0.976	0.951	0.951	0.951	0.951	0.951	0.935
12	nli-mpnet-base-v2	0.73	0.986	0.976	0.957	0.945	0.951	0.954	0.947	0.935
13	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.73	0.992	0.975	0.974	0.920	0.946	0.963	0.931	0.930
14	AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1	0.67	0.996	0.975	0.987	0.908	0.946	0.970	0.923	0.931
15	distiluse-base-multilingual-cased-v2	0.64	0.982	0.973	0.945	0.945	0.945	0.945	0.945	0.927
16	distiluse-base-multilingual-cased	0.64	0.982	0.973	0.945	0.945	0.945	0.945	0.945	0.927
17	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.73	0.992	0.973	0.974	0.914	0.943	0.961	0.925	0.926
18	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.73	0.992	0.973	0.974	0.914	0.943	0.961	0.925	0.926
19	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.73	0.992	0.973	0.974	0.914	0.943	0.961	0.925	0.926
20	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.73	0.992	0.973	0.974	0.914	0.943	0.961	0.925	0.926
21	paraphrase-multilingual-MiniLM-L12-v2	0.73	0.992	0.973	0.974	0.914	0.943	0.961	0.925	0.926
22	bert-large-nli-max-tokens	0.86	0.994	0.973	0.980	0.908	0.943	0.965	0.922	0.926
23	nli-bert-large-max-pooling	0.86	0.994	0.973	0.980	0.908	0.943	0.965	0.922	0.926
24	bert-large-nli-stsb-mean-tokens	0.69	0.990	0.972	0.968	0.914	0.940	0.956	0.924	0.922
25	multi-qa-distilbert-dot-v1	0.61	0.990	0.972	0.968	0.914	0.940	0.956	0.924	0.922
26	distilbert-base-nli-stsb-wkpooling	0.71	0.992	0.972	0.974	0.908	0.940	0.960	0.920	0.922
27	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.71	0.992	0.972	0.974	0.908	0.940	0.960	0.920	0.922
28	paraphrase-distilroberta-base-v2	0.67	0.982	0.970	0.944	0.933	0.938	0.942	0.935	0.919
29	nli-roberta-large	0.84	0.996	0.970	0.986	0.890	0.935	0.965	0.907	0.918
30	paraphrase-TinyBERT-L6-v2	0.67	0.988	0.969	0.961	0.908	0.934	0.950	0.918	0.914
31	nli-roberta-base-v2	0.77	0.990	0.969	0.967	0.902	0.933	0.953	0.914	0.914
32	multi-qa-mpnet-base-dot-v1	0.70	0.980	0.967	0.938	0.926	0.932	0.936	0.929	0.910

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
33	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs	0.74	0.994	0.969	0.980	0.890	0.932	0.960	0.906	0.914
34	multi-qa-MiniLM-L6-dot-v1	0.77	0.974	0.966	0.922	0.939	0.930	0.925	0.935	0.907
35	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot	0.73	0.988	0.967	0.961	0.902	0.930	0.948	0.913	0.910
36	paraphrase-MiniLM-L12-v2	0.68	0.988	0.967	0.961	0.902	0.930	0.948	0.913	0.910
37	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs	0.74	0.990	0.967	0.967	0.896	0.930	0.952	0.909	0.910
38	bert-large-nli-cls-token	0.87	0.992	0.967	0.973	0.890	0.929	0.955	0.905	0.910
39	roberta-base-nli-stsb-mean-tokens	0.70	0.992	0.967	0.973	0.890	0.929	0.955	0.905	0.910
40	nli-bert-large-cls-pooling	0.87	0.992	0.967	0.973	0.890	0.929	0.955	0.905	0.910
41	nli-distilroberta-base-v2	0.73	0.982	0.966	0.943	0.914	0.928	0.937	0.920	0.906
42	distilbert-base-nli-stsb-mean-tokens	0.71	0.990	0.966	0.967	0.890	0.927	0.950	0.904	0.906
43	all-MiniLM-L12-v1	0.66	0.970	0.963	0.911	0.939	0.924	0.916	0.933	0.900
44	bert-base-nli-stsb-mean-tokens	0.73	0.986	0.964	0.954	0.896	0.924	0.942	0.907	0.901
45	roberta-large-nli-mean-tokens	0.83	0.988	0.964	0.960	0.890	0.924	0.945	0.903	0.901
46	paraphrase-albert-base-v2	0.68	0.976	0.963	0.926	0.920	0.923	0.925	0.921	0.898
47	DataikuNLP_paraphrase-MiniLM-L6-v2	0.67	0.980	0.963	0.937	0.908	0.922	0.931	0.914	0.898
48	paraphrase-MiniLM-L6-v2	0.67	0.980	0.963	0.937	0.908	0.922	0.931	0.914	0.898
49	AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2	0.63	0.984	0.963	0.948	0.896	0.921	0.937	0.906	0.897
50	msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.70	0.984	0.963	0.948	0.896	0.921	0.937	0.906	0.897
51	distilroberta-base-paraphrase-v1	0.65	0.986	0.963	0.954	0.890	0.921	0.940	0.902	0.897
52	paraphrase-distilroberta-base-v1	0.65	0.986	0.963	0.954	0.890	0.921	0.940	0.902	0.897
53	all-MiniLM-L6-v2	0.66	0.976	0.961	0.925	0.914	0.920	0.923	0.916	0.894
54	deutsche-telekom_gbert-large-paraphrase-cosine	0.71	0.976	0.961	0.925	0.914	0.920	0.923	0.916	0.894
55	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot	0.72	0.978	0.961	0.931	0.908	0.919	0.926	0.912	0.894
56	deutsche-telekom_gbert-large-paraphrase-euclidean	0.76	0.980	0.961	0.936	0.902	0.919	0.929	0.909	0.894
57	all-MiniLM-L12-v2	0.65	0.968	0.960	0.905	0.933	0.918	0.910	0.927	0.892
58	msmarco-bert-base-dot-v5	0.93	0.972	0.960	0.915	0.920	0.917	0.916	0.919	0.891
59	Hoax0930_paraphrase-multilingual-mpnet-base-v2	0.78	0.988	0.961	0.960	0.877	0.917	0.942	0.893	0.893
60	all-MiniLM-L6-v1	0.69	0.988	0.961	0.960	0.877	0.917	0.942	0.893	0.893

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
61	bert-base-nli-cls-token	0.84	0.982	0.960	0.942	0.890	0.915	0.931	0.900	0.889
62	facebook-dpr-question_encoder-single-nq-base	0.91	0.982	0.960	0.942	0.890	0.915	0.931	0.900	0.889
63	nli-bert-base-cls-pooling	0.84	0.982	0.960	0.942	0.890	0.915	0.931	0.900	0.889
64	bert-base-nli-mean-tokens	0.80	0.972	0.958	0.914	0.914	0.914	0.914	0.914	0.886
65	bert-base-nli-wkpooling	0.80	0.972	0.958	0.914	0.914	0.914	0.914	0.914	0.886
66	nli-bert-base	0.80	0.972	0.958	0.914	0.914	0.914	0.914	0.914	0.886
67	DataikuNLP_paraphrase-albert-small-v2	0.64	0.974	0.958	0.919	0.908	0.914	0.917	0.910	0.886
68	paraphrase-albert-small-v2	0.64	0.974	0.958	0.919	0.908	0.914	0.917	0.910	0.886
69	distilbert-base-nli-stsb-quora-ranking	0.71	0.980	0.958	0.935	0.890	0.912	0.926	0.898	0.885
70	quora-distilbert-base	0.71	0.980	0.958	0.935	0.890	0.912	0.926	0.898	0.885
71	msmarco-distilbert-base-tas-b	0.87	0.982	0.958	0.941	0.883	0.911	0.929	0.894	0.885
72	facebook-dpr-question_encoder-multiset-base	0.87	0.966	0.955	0.898	0.920	0.909	0.903	0.916	0.880
73	msmarco-distilbert-base-v3	0.59	0.980	0.957	0.935	0.883	0.909	0.924	0.893	0.881
74	msmarco-bert-co-condensor	0.93	0.970	0.955	0.908	0.908	0.908	0.908	0.908	0.878
75	bert-large-nli-mean-tokens	0.84	0.986	0.957	0.953	0.865	0.907	0.934	0.881	0.880
76	nli-bert-large	0.84	0.986	0.957	0.953	0.865	0.907	0.934	0.881	0.880
77	paraphrase-MiniLM-L3-v2	0.62	0.966	0.954	0.898	0.914	0.906	0.901	0.911	0.875
78	all-mpnet-base-v1	0.67	0.976	0.955	0.924	0.890	0.906	0.917	0.896	0.877
79	distilbert-base-nli-wkpooling	0.83	0.976	0.955	0.924	0.890	0.906	0.917	0.896	0.877
80	multi-qa-MiniLM-L6-cos-v1	0.67	0.980	0.955	0.935	0.877	0.905	0.923	0.888	0.877
81	msmarco-distilbert-cos-v5	0.60	0.974	0.954	0.918	0.890	0.903	0.912	0.895	0.873
82	msmarco-distilbert-base-v4	0.60	0.974	0.954	0.918	0.890	0.903	0.912	0.895	0.873
83	Hoax0930_paraphrase-multilingual-MiniLM-L12-v2	0.73	0.966	0.952	0.897	0.908	0.902	0.899	0.906	0.871
84	multi-qa-distilbert-cos-v1	0.65	0.966	0.952	0.897	0.908	0.902	0.899	0.906	0.871
85	msmarco-distilbert-base-dot-prod-v3	0.75	0.968	0.952	0.902	0.902	0.902	0.902	0.902	0.870
86	multi-qa-mpnet-base-cos-v1	0.67	0.972	0.952	0.912	0.890	0.901	0.907	0.894	0.869
87	nli-roberta-base	0.83	0.974	0.952	0.917	0.883	0.900	0.910	0.890	0.869
88	facebook-dpr-ctx_encoder-multiset-base	0.86	0.986	0.954	0.952	0.853	0.900	0.930	0.871	0.872
89	roberta-base-nli-mean-tokens	0.84	0.984	0.952	0.946	0.853	0.897	0.925	0.870	0.868

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
90	distilbert-base-nli-max-tokens	0.89	0.986	0.952	0.952	0.847	0.896	0.929	0.866	0.868
91	distilbert-base-nli-mean-tokens	0.81	0.978	0.951	0.928	0.865	0.895	0.914	0.877	0.864
92	nli-distilbert-base	0.81	0.978	0.951	0.928	0.865	0.895	0.914	0.877	0.864
93	msmarco-roberta-base-v2	0.66	0.974	0.949	0.916	0.871	0.893	0.907	0.880	0.860
94	bert-base-nli-max-tokens	0.84	0.986	0.951	0.951	0.840	0.893	0.927	0.861	0.864
95	nli-bert-base-max-pooling	0.84	0.986	0.951	0.951	0.840	0.893	0.927	0.861	0.864
96	all-distilroberta-v1	0.65	0.959	0.946	0.876	0.908	0.892	0.882	0.901	0.856
97	all-mpnet-base-v2	0.68	0.966	0.948	0.895	0.890	0.892	0.894	0.891	0.858
98	JoBeer_paraphrase-MiniLM-L6-v2-eclass	0.74	0.974	0.948	0.916	0.865	0.890	0.905	0.875	0.856
99	gart-labor_paraphrase-MiniLM-L6-v2-eclass	0.74	0.974	0.948	0.916	0.865	0.890	0.905	0.875	0.856
100	distilbert-multilingual-nli-stsb-quora-ranking	0.95	0.984	0.949	0.945	0.840	0.890	0.922	0.859	0.859
101	quora-distilbert-multilingual	0.95	0.984	0.949	0.945	0.840	0.890	0.922	0.859	0.859
102	nq-distilbert-base-v1	0.57	0.978	0.948	0.927	0.853	0.888	0.911	0.867	0.855
103	msmarco-roberta-base-v3	0.56	0.970	0.946	0.904	0.871	0.887	0.898	0.878	0.852
104	JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass	0.74	0.972	0.946	0.910	0.865	0.887	0.900	0.874	0.852
105	all-roberta-large-v1	0.68	0.966	0.945	0.894	0.877	0.885	0.890	0.881	0.849
106	msmarco-MiniLM-L12-cos-v5	0.61	0.976	0.946	0.921	0.853	0.885	0.906	0.866	0.851
107	msmarco-MiniLM-L-12-v3	0.61	0.976	0.946	0.921	0.853	0.885	0.906	0.866	0.851
108	msmarco-distilbert-dot-v5	0.85	0.976	0.946	0.921	0.853	0.885	0.906	0.866	0.851
109	new5558_chula-course-paraphrase-multilingual-mpnet-base-v2	0.75	0.970	0.945	0.904	0.865	0.884	0.896	0.873	0.848
110	distilroberta-base-msmarco-v2	0.61	0.959	0.940	0.873	0.883	0.878	0.875	0.881	0.839
111	msmarco-distilroberta-base-v2	0.61	0.959	0.940	0.873	0.883	0.878	0.875	0.881	0.839
112	msmarco-distilbert-base-v2	0.65	0.980	0.943	0.931	0.828	0.877	0.908	0.847	0.842
113	msmarco-MiniLM-L-6-v3	0.60	0.976	0.942	0.919	0.834	0.875	0.901	0.850	0.838
114	msmarco-MiniLM-L6-cos-v5	0.60	0.976	0.942	0.919	0.834	0.875	0.901	0.850	0.838
115	nli-distilbert-base-max-pooling	0.78	0.968	0.939	0.896	0.847	0.871	0.886	0.856	0.831
116	ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8	0.82	0.961	0.937	0.876	0.865	0.870	0.874	0.867	0.829
117	bert-base-nli-stsb-wkpooling	0.76	0.972	0.939	0.907	0.834	0.869	0.891	0.848	0.830
118	distilroberta-base-msmarco-v1	0.71	0.966	0.937	0.890	0.847	0.868	0.881	0.855	0.827

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
119	msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.62	0.976	0.937	0.917	0.816	0.864	0.895	0.834	0.825
120	gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.74	0.955	0.931	0.859	0.859	0.859	0.859	0.859	0.814
121	hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot	0.75	0.951	0.930	0.849	0.865	0.857	0.852	0.862	0.811
122	facebook-dpr-ctx_encoder-single-nq-base	0.82	0.963	0.928	0.876	0.822	0.848	0.865	0.832	0.802
123	average_word_embeddings_komninos	0.88	0.947	0.915	0.831	0.816	0.824	0.828	0.819	0.768
124	Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2	0.42	0.923	0.906	0.781	0.853	0.815	0.794	0.837	0.754
125	JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp	0.77	0.959	0.913	0.857	0.773	0.813	0.839	0.788	0.758
126	allenai-specter	0.90	0.959	0.913	0.857	0.773	0.813	0.839	0.788	0.758
127	average_word_embeddings_glove.6B.300d	0.82	0.955	0.909	0.845	0.767	0.804	0.828	0.781	0.746
128	gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.76	0.921	0.897	0.770	0.822	0.795	0.780	0.811	0.727
129	average_word_embeddings_levy_dependency	0.91	0.951	0.904	0.832	0.761	0.795	0.817	0.774	0.734
130	average_word_embeddings_glove.840B.300d	0.85	0.945	0.896	0.812	0.742	0.776	0.797	0.755	0.709
131	Prompsit_paraphrase-bert-en	0.93	0.913	0.870	0.732	0.736	0.734	0.733	0.735	0.648
132	Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2	0.32	0.905	0.855	0.704	0.699	0.702	0.703	0.700	0.606
133	cointegrated_rubert-base-cased-dp-paraphrase-detection	0.92	0.949	0.867	0.794	0.613	0.692	0.750	0.643	0.617
134	valurank_paraphrase-mpnet-base-v2-offensive	0.99	0.947	0.864	0.786	0.607	0.685	0.742	0.636	0.608
135	orenpereg_paraphrase-mpnet-base-v2_sst2_64samps	0.90	0.892	0.833	0.658	0.650	0.654	0.657	0.652	0.544
136	clip-ViT-B-32-multilingual-v1	0.97	0.714	0.725	0.461	0.761	0.574	0.500	0.673	0.416
137	msmarco-roberta-base-ance-firstp	0.99	0.505	0.625	0.394	1.000	0.565	0.448	0.765	0.446
138	msmarco-roberta-base-ance-fristp	0.99	0.505	0.625	0.394	1.000	0.565	0.448	0.765	0.446
139	moshew_paraphrase-mpnet-base-v2_SetFit_emotions	0.87	0.801	0.751	0.490	0.595	0.537	0.508	0.571	0.372
140	bert-base-wikipedia-sections-mean-tokens	0.99	0.432	0.567	0.359	0.988	0.526	0.411	0.731	0.383
141	moshew_paraphrase-mpnet-base-v2_SetFit_sst2	0.99	0.740	0.684	0.386	0.509	0.439	0.406	0.479	0.229
142	aditeyabaral-bert-base-cased	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
143	aditeyabaral-contrastive-roberta-base	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
144	aditeyabaral-distilbert-base-cased	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
145	aditeyabaral-roberta-base	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000
146	aditeyabaral-xlm-roberta-base	0.00	0.000	0.243	0.243	1.000	0.391	0.287	0.616	0.000

Tablica 37. Rezultati jezičnih modela drugog ciklusa eksperimenata za korpus VMEN

#	Model	*θ*	*S*	*A*	*P*	*R*	F1	F.5	F2	MCC
1	deutsche-telekom_gbert-large-paraphrase-cosine	0.89	1.000	0.999	1.000	0.970	0.984	0.994	0.976	0.984
2	average_word_embeddings_komninos	0.97	0.999	0.999	0.981	0.980	0.980	0.981	0.980	0.980
3	average_word_embeddings_levy_dependency	0.98	1.000	0.999	0.990	0.970	0.980	0.986	0.974	0.979
4	deutsche-telekom_gbert-large-paraphrase-euclidean	0.92	1.000	0.999	1.000	0.950	0.973	0.989	0.959	0.974
5	LaBSE	0.88	1.000	0.998	0.990	0.950	0.969	0.981	0.957	0.969
6	distiluse-base-multilingual-cased-v2	0.86	1.000	0.998	0.989	0.940	0.964	0.979	0.949	0.963
7	distiluse-base-multilingual-cased	0.86	1.000	0.998	0.989	0.940	0.964	0.979	0.949	0.963
8	paraphrase-TinyBERT-L6-v2	0.72	1.000	0.998	0.990	0.940	0.963	0.978	0.949	0.963
9	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.79	1.000	0.998	0.990	0.930	0.958	0.976	0.940	0.958
10	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.79	1.000	0.998	0.990	0.930	0.958	0.976	0.940	0.958
11	paraphrase-multilingual-MiniLM-L12-v2	0.79	1.000	0.998	0.990	0.930	0.958	0.976	0.940	0.958
12	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.83	1.000	0.998	1.000	0.920	0.957	0.982	0.934	0.958
13	bert-large-nli-cls-token	0.94	1.000	0.998	1.000	0.920	0.957	0.982	0.934	0.957
14	nli-bert-large-cls-pooling	0.94	1.000	0.998	1.000	0.920	0.957	0.982	0.934	0.957
15	nli-distilroberta-base-v2	0.91	0.999	0.998	0.981	0.930	0.954	0.970	0.939	0.953
16	multi-qa-distilbert-dot-v1	0.72	0.999	0.998	0.971	0.940	0.953	0.963	0.945	0.953
17	bert-large-nli-max-tokens	0.94	1.000	0.998	1.000	0.910	0.952	0.980	0.926	0.952
18	bert-large-nli-mean-tokens	0.93	1.000	0.998	0.989	0.920	0.952	0.974	0.932	0.952
19	nli-bert-large-max-pooling	0.94	1.000	0.998	1.000	0.910	0.952	0.980	0.926	0.952
20	nli-bert-large	0.93	1.000	0.998	0.989	0.920	0.952	0.974	0.932	0.952
21	msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.83	1.000	0.998	1.000	0.910	0.951	0.979	0.925	0.952
22	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.83	1.000	0.998	1.000	0.910	0.951	0.980	0.926	0.952
23	bert-base-nli-cls-token	0.95	0.999	0.997	0.980	0.920	0.948	0.967	0.931	0.948
24	nli-bert-base-cls-pooling	0.95	0.999	0.997	0.980	0.920	0.948	0.967	0.931	0.948
25	nli-roberta-base-v2	0.92	0.999	0.997	0.981	0.920	0.948	0.967	0.931	0.948
26	multi-qa-mpnet-base-dot-v1	0.80	0.999	0.997	0.980	0.920	0.947	0.966	0.930	0.947
27	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.76	1.000	0.997	0.988	0.910	0.947	0.971	0.924	0.947
28	paraphrase-distilroberta-base-v2	0.79	1.000	0.997	1.000	0.900	0.947	0.978	0.918	0.947

29	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.75	1.000	0.997	0.990	0.910	0.946	0.972	0.924	0.947
30	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.83	1.000	0.998	1.000	0.900	0.945	0.977	0.917	0.946
31	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.80	1.000	0.998	1.000	0.900	0.945	0.977	0.917	0.946
32	paraphrase-multilingual-mpnet-base-v2	0.80	1.000	0.998	1.000	0.900	0.945	0.977	0.917	0.946
33	AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1	0.80	1.000	0.997	1.000	0.900	0.945	0.977	0.917	0.946
34	paraphrase-mpnet-base-v2	0.79	1.000	0.998	1.000	0.900	0.944	0.976	0.916	0.946
35	bert-base-nli-max-tokens	0.94	0.999	0.997	0.980	0.910	0.941	0.964	0.922	0.942
36	nli-bert-base-max-pooling	0.94	0.999	0.997	0.980	0.910	0.941	0.964	0.922	0.942
37	multi-qa-distilbert-cos-v1	0.77	0.999	0.997	0.981	0.910	0.941	0.964	0.922	0.942
38	multi-qa-mpnet-base-cos-v1	0.79	1.000	0.997	0.989	0.900	0.941	0.969	0.916	0.941
39	DataikuNLP_paraphrase-albert-small-v2	0.71	1.000	0.997	1.000	0.890	0.941	0.975	0.909	0.941
40	paraphrase-albert-small-v2	0.71	1.000	0.997	1.000	0.890	0.941	0.975	0.909	0.941
41	nli-mpnet-base-v2	0.91	1.000	0.997	0.989	0.900	0.941	0.969	0.916	0.942
42	facebook-dpr-ctx_encoder-single-nq-base	0.91	1.000	0.997	0.988	0.900	0.940	0.968	0.915	0.941
43	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.70	1.000	0.997	1.000	0.890	0.940	0.975	0.909	0.941
44	distilbert-base-nli-stsb-mean-tokens	0.83	1.000	0.997	1.000	0.890	0.939	0.974	0.909	0.941
45	distilbert-multilingual-nli-stsb-quora-ranking	0.97	1.000	0.997	1.000	0.890	0.939	0.974	0.909	0.941
46	quora-distilbert-multilingual	0.97	1.000	0.997	1.000	0.890	0.939	0.974	0.909	0.941
47	distilbert-base-nli-stsb-wkpooling	0.87	0.999	0.997	0.980	0.900	0.937	0.962	0.914	0.937
48	facebook-dpr-question_encoder-multiset-base	0.95	1.000	0.997	0.989	0.890	0.936	0.967	0.908	0.936
49	distilroberta-base-paraphrase-v1	0.75	1.000	0.997	1.000	0.880	0.935	0.973	0.901	0.936
50	facebook-dpr-question_encoder-single-nq-base	0.97	1.000	0.997	1.000	0.880	0.935	0.973	0.901	0.936
51	paraphrase-distilroberta-base-v1	0.75	1.000	0.997	1.000	0.880	0.935	0.973	0.901	0.936
52	all-MiniLM-L6-v2	0.82	1.000	0.997	0.989	0.890	0.935	0.966	0.907	0.936
53	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs	0.83	1.000	0.997	0.990	0.890	0.935	0.966	0.907	0.936
54	paraphrase-MiniLM-L12-v2	0.77	1.000	0.997	1.000	0.880	0.934	0.972	0.900	0.935
55	bert-large-nli-stsb-mean-tokens	0.85	0.999	0.996	0.963	0.910	0.933	0.950	0.918	0.933
56	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot	0.81	1.000	0.997	1.000	0.880	0.933	0.971	0.900	0.935
57	nli-distilbert-base-max-pooling	0.92	0.999	0.997	0.968	0.900	0.931	0.952	0.912	0.931
58	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.80	1.000	0.997	0.990	0.890	0.931	0.964	0.905	0.934

59	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.80	1.000	0.997	0.990	0.890	0.931	0.964	0.905	0.934
60	multi-qa-MiniLM-L6-cos-v1	0.80	0.999	0.997	0.979	0.890	0.930	0.958	0.905	0.931
61	nli-roberta-large	0.91	0.999	0.997	0.980	0.890	0.930	0.959	0.905	0.931
62	nli-distilbert-base	0.93	1.000	0.997	0.988	0.880	0.930	0.964	0.899	0.930
63	msmarco-distilbert-base-v2	0.80	1.000	0.996	1.000	0.870	0.930	0.970	0.893	0.931
64	paraphrase-albert-base-v2	0.75	1.000	0.997	1.000	0.870	0.929	0.970	0.892	0.930
65	roberta-base-nli-mean-tokens	0.91	1.000	0.997	0.988	0.880	0.929	0.963	0.899	0.930
66	roberta-large-nli-stsb-mean-tokens	0.80	0.999	0.996	0.970	0.890	0.927	0.952	0.904	0.927
67	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.79	1.000	0.997	0.989	0.880	0.926	0.961	0.897	0.929
68	roberta-base-nli-stsb-mean-tokens	0.85	0.999	0.996	0.979	0.880	0.925	0.956	0.897	0.926
69	bert-base-nli-stsb-mean-tokens	0.90	1.000	0.996	1.000	0.860	0.924	0.968	0.884	0.925
70	bert-base-nli-mean-tokens	0.92	1.000	0.996	0.978	0.880	0.924	0.955	0.896	0.925
71	bert-base-nli-wkpooling	0.92	1.000	0.996	0.978	0.880	0.924	0.955	0.896	0.925
72	nli-bert-base	0.92	1.000	0.996	0.978	0.880	0.924	0.955	0.896	0.925
73	all-MiniLM-L6-v1	0.79	1.000	0.996	1.000	0.860	0.924	0.968	0.884	0.925
74	average_word_embeddings_glove.840B.300d	0.97	1.000	0.997	1.000	0.860	0.923	0.967	0.884	0.925
75	msmarco-roberta-base-v2	0.82	0.999	0.996	0.980	0.870	0.921	0.955	0.889	0.921
76	roberta-large-nli-mean-tokens	0.90	0.999	0.996	0.978	0.870	0.920	0.954	0.889	0.920
77	nli-roberta-base	0.94	0.999	0.996	0.969	0.880	0.920	0.948	0.895	0.920
78	msmarco-bert-base-dot-v5	0.96	1.000	0.996	1.000	0.860	0.920	0.965	0.883	0.924
79	distilbert-base-nli-wkpooling	0.92	0.999	0.996	0.978	0.870	0.919	0.953	0.889	0.920
80	AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2	0.74	1.000	0.996	0.989	0.860	0.918	0.959	0.882	0.919
81	msmarco-distilbert-base-dot-prod-v3	0.89	1.000	0.996	0.988	0.860	0.918	0.958	0.882	0.919
82	distilbert-base-nli-max-tokens	0.94	1.000	0.996	0.988	0.860	0.918	0.959	0.882	0.919
83	multi-qa-MiniLM-L6-dot-v1	0.86	0.999	0.996	0.970	0.880	0.917	0.946	0.893	0.919
84	all-mpnet-base-v2	0.88	0.999	0.996	0.980	0.870	0.917	0.953	0.888	0.919
85	msmarco-distilbert-base-v4	0.77	1.000	0.996	0.988	0.860	0.917	0.958	0.882	0.919
86	distiluse-base-multilingual-cased-v1	0.77	1.000	0.996	0.990	0.860	0.917	0.958	0.881	0.919
87	average_word_embeddings_glove.6B.300d	0.97	1.000	0.996	1.000	0.850	0.917	0.964	0.875	0.919
88	paraphrase-MiniLM-L3-v2	0.72	1.000	0.996	0.979	0.860	0.914	0.951	0.880	0.914

89	JoBeer_paraphrase-MiniLM-L6-v2-eclass	0.78	0.999	0.996	0.978	0.860	0.912	0.949	0.879	0.913
90	gart-labor_paraphrase-MiniLM-L6-v2-eclass	0.78	0.999	0.996	0.978	0.860	0.912	0.949	0.879	0.913
91	msmarco-distilbert-cos-v5	0.77	1.000	0.996	0.978	0.860	0.912	0.949	0.880	0.913
92	msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.77	1.000	0.996	1.000	0.840	0.911	0.961	0.866	0.913
93	DataikuNLP_paraphrase-MiniLM-L6-v2	0.80	1.000	0.996	1.000	0.840	0.911	0.962	0.866	0.913
94	paraphrase-MiniLM-L6-v2	0.80	1.000	0.996	1.000	0.840	0.911	0.962	0.866	0.913
95	msmarco-MiniLM-L-6-v3	0.76	0.999	0.996	0.981	0.850	0.907	0.948	0.871	0.909
96	Hoax0930_paraphrase-multilingual-MiniLM-L12-v2	0.85	0.999	0.995	0.964	0.860	0.907	0.939	0.877	0.907
97	msmarco-distilbert-dot-v5	0.91	1.000	0.995	0.990	0.840	0.906	0.954	0.865	0.908
98	msmarco-distilbert-base-tas-b	0.93	1.000	0.996	0.990	0.840	0.905	0.954	0.864	0.908
99	msmarco-bert-co-condensor	0.96	0.999	0.995	0.966	0.850	0.903	0.940	0.870	0.903
100	nq-distilbert-base-v1	0.75	1.000	0.995	0.990	0.840	0.903	0.951	0.863	0.907
101	allenai-specter	0.92	0.999	0.995	0.976	0.840	0.902	0.945	0.863	0.903
102	msmarco-MiniLM-L6-cos-v5	0.76	0.999	0.995	0.980	0.840	0.901	0.946	0.863	0.903
103	bert-base-nli-stsb-wkpooling	0.87	0.999	0.995	0.957	0.850	0.899	0.933	0.869	0.899
104	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs	0.84	0.999	0.995	0.965	0.850	0.896	0.934	0.866	0.899
105	msmarco-distilbert-base-v3	0.81	1.000	0.995	1.000	0.810	0.894	0.954	0.841	0.897
106	distilroberta-base-msmarco-v2	0.83	0.999	0.995	0.971	0.830	0.892	0.937	0.853	0.894
107	msmarco-distilroberta-base-v2	0.83	0.999	0.995	0.971	0.830	0.892	0.937	0.853	0.894
108	all-roberta-large-v1	0.83	0.999	0.995	0.978	0.820	0.891	0.941	0.846	0.892
109	all-mpnet-base-v1	0.82	0.999	0.995	0.979	0.820	0.891	0.941	0.846	0.893
110	Hoax0930_paraphrase-multilingual-mpnet-base-v2	0.87	1.000	0.995	0.976	0.820	0.891	0.940	0.847	0.892
111	distilbert-base-nli-mean-tokens	0.90	0.999	0.995	0.959	0.840	0.890	0.928	0.858	0.892
112	msmarco-MiniLM-L-12-v3	0.81	0.999	0.995	0.980	0.820	0.889	0.940	0.845	0.892
113	msmarco-roberta-base-v3	0.78	0.999	0.995	0.981	0.820	0.888	0.940	0.845	0.892
114	msmarco-MiniLM-L12-cos-v5	0.81	1.000	0.995	0.989	0.810	0.888	0.945	0.839	0.891
115	orenpereg_paraphrase-mpnet-base-v2_sst2_64samps	0.92	0.999	0.995	0.980	0.810	0.884	0.938	0.837	0.887
116	quora-distilbert-base	0.79	0.999	0.995	0.980	0.810	0.883	0.937	0.837	0.886
117	distilroberta-base-msmarco-v1	0.85	1.000	0.995	0.988	0.800	0.883	0.943	0.831	0.886

118	JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass	0.76	1.000	0.995	0.975	0.810	0.882	0.935	0.837	0.885
119	all-MiniLM-L12-v1	0.82	0.999	0.994	0.978	0.800	0.878	0.935	0.829	0.881
120	all-MiniLM-L12-v2	0.83	0.999	0.994	0.978	0.780	0.865	0.929	0.811	0.869
121	facebook-dpr-ctx_encoder-multiset-base	0.94	1.000	0.994	1.000	0.750	0.852	0.933	0.787	0.861
122	all-distilroberta-v1	0.81	0.999	0.993	0.946	0.780	0.850	0.903	0.805	0.853
123	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot	0.84	0.999	0.993	0.960	0.770	0.845	0.907	0.797	0.852
124	distilbert-base-nli-stsb-quora-ranking	0.86	0.999	0.993	0.952	0.760	0.841	0.903	0.790	0.845
125	new5558_chula-course-paraphrase-multilingual-mpnet-base-v2	0.82	0.999	0.993	0.962	0.750	0.837	0.906	0.782	0.843
126	valurank_paraphrase-mpnet-base-v2-offensive	0.99	0.995	0.991	0.821	0.840	0.829	0.824	0.835	0.825
127	JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp	0.85	1.000	0.991	0.987	0.670	0.789	0.893	0.712	0.805
128	ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8	0.94	1.000	0.991	1.000	0.650	0.787	0.902	0.699	0.802
129	Prompsit_paraphrase-bert-en	0.98	1.000	0.991	1.000	0.650	0.786	0.901	0.698	0.802
130	gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.83	0.997	0.989	0.885	0.720	0.772	0.829	0.736	0.783
131	hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot	0.91	0.999	0.990	0.953	0.650	0.763	0.862	0.689	0.777
132	cointegrated_rubert-base-cased-dp-paraphrase-detection	0.98	1.000	0.989	1.000	0.580	0.726	0.863	0.630	0.753
133	gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.93	0.999	0.988	0.919	0.600	0.717	0.822	0.640	0.733
134	Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2	0.67	0.998	0.984	0.921	0.450	0.590	0.743	0.496	0.629
135	Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2	0.73	0.997	0.979	0.798	0.280	0.401	0.556	0.318	0.454
136	moshew_paraphrase-mpnet-base-v2_SetFit_emotions	0.98	0.982	0.967	0.478	0.410	0.377	0.410	0.384	0.394
137	bert-base-wikipedia-sections-mean-tokens	0.99	0.861	0.864	0.163	0.990	0.279	0.195	0.488	0.371
138	moshew_paraphrase-mpnet-base-v2_SetFit_sst2	0.99	0.747	0.746	0.077	0.710	0.137	0.093	0.262	0.173
139	msmarco-roberta-base-ance-firstp	0.99	0.542	0.553	0.055	1.000	0.105	0.068	0.226	0.173
140	msmarco-roberta-base-ance-fristp	0.99	0.542	0.553	0.055	1.000	0.105	0.068	0.226	0.173
141	clip-ViT-B-32-multilingual-v1	0.99	0.218	0.238	0.033	0.990	0.064	0.041	0.145	0.080
142	aditeyabaral-bert-base-cased	0.00	0.000	0.026	0.026	1.000	0.050	0.032	0.116	0.000
143	aditeyabaral-contrastive-roberta-base	0.00	0.000	0.026	0.026	1.000	0.050	0.032	0.116	0.000
144	aditeyabaral-distilbert-base-cased	0.00	0.000	0.026	0.026	1.000	0.050	0.032	0.116	0.000
145	aditeyabaral-roberta-base	0.00	0.000	0.026	0.026	1.000	0.050	0.032	0.116	0.000
146	aditeyabaral-xlm-roberta-base	0.00	0.000	0.026	0.026	1.000	0.050	0.032	0.116	0.000

Tablica 38. Rezultati jezičnih modela drugog ciklusa eksperimenata za korpus Webis

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
1	AIDA-UPM_mstsb-paraphrase-multilingual-mpnet-base-v2	0.35	0.126	0.566	0.538	0.996	0.699	0.593	0.851	0.249
2	all-mpnet-base-v2	0.38	0.116	0.563	0.536	1.000	0.698	0.591	0.852	0.249
3	all-mpnet-base-v1	0.28	0.117	0.563	0.536	0.999	0.698	0.591	0.852	0.247
4	deutsche-telekom_gbert-large-paraphrase-cosine	0.43	0.117	0.563	0.536	0.999	0.698	0.591	0.852	0.247
5	DataikuNLP_paraphrase-multilingual-MiniLM-L12-v2	0.42	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
6	Hoax0930_pseudo_paraphrase-multilingual-MiniLM-L12-v2	0.42	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
7	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-1shot	0.37	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
8	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_10_Epochs	0.43	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
9	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_1_Epochs	0.43	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
10	keithhon_paraphrase-multilingual-MiniLM-L12-v2	0.42	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
11	paraphrase-multilingual-MiniLM-L12-v2	0.42	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
12	roberta-base-nli-mean-tokens	0.68	0.124	0.564	0.537	0.995	0.698	0.591	0.850	0.243
13	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-8shot	0.43	0.125	0.565	0.538	0.996	0.698	0.592	0.851	0.248
14	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_100_Epochs	0.45	0.125	0.565	0.537	0.995	0.698	0.592	0.850	0.244
15	paraphrase-distilroberta-base-v2	0.40	0.125	0.565	0.538	0.996	0.698	0.592	0.851	0.248
16	jfarray_Model_paraphrase-multilingual-MiniLM-L12-v2_50_Epochs	0.45	0.126	0.565	0.538	0.995	0.698	0.592	0.850	0.246
17	distilbert-base-nli-stsb-mean-tokens	0.50	0.127	0.566	0.538	0.995	0.698	0.593	0.851	0.247
18	roberta-base-nli-stsb-mean-tokens	0.47	0.127	0.565	0.538	0.994	0.698	0.592	0.850	0.244
19	roberta-large-nli-mean-tokens	0.68	0.127	0.566	0.538	0.995	0.698	0.593	0.851	0.247
20	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_1_Epochs	0.52	0.129	0.565	0.538	0.992	0.698	0.592	0.849	0.241
21	paraphrase-multilingual-mpnet-base-v2	0.52	0.129	0.565	0.538	0.992	0.698	0.592	0.849	0.241
22	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_50_Epochs	0.57	0.131	0.566	0.538	0.991	0.698	0.592	0.848	0.241
23	roberta-large-nli-stsb-mean-tokens	0.50	0.131	0.567	0.539	0.992	0.698	0.593	0.849	0.244
24	bert-base-nli-stsb-mean-tokens	0.44	0.111	0.560	0.535	1.000	0.697	0.590	0.852	0.243

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
25	distilbert-base-nli-stsb-wkpooling	0.51	0.111	0.560	0.535	1.000	0.697	0.590	0.852	0.243
26	distilbert-base-nli-wkpooling	0.70	0.111	0.560	0.535	1.000	0.697	0.590	0.852	0.243
27	nli-roberta-base	0.69	0.112	0.561	0.535	1.000	0.697	0.590	0.852	0.245
28	msmarco-bert-co-condensor	0.87	0.113	0.561	0.535	1.000	0.697	0.590	0.852	0.246
29	nli-roberta-base-v2	0.45	0.113	0.561	0.535	0.999	0.697	0.590	0.851	0.242
30	nli-distilroberta-base-v2	0.47	0.115	0.561	0.535	0.999	0.697	0.590	0.851	0.244
31	nli-roberta-large	0.71	0.115	0.561	0.535	0.999	0.697	0.590	0.851	0.244
32	all-MiniLM-L6-v1	0.24	0.116	0.561	0.535	0.997	0.697	0.590	0.851	0.241
33	deutsche-telekom_gbert-large-paraphrase-euclidean	0.59	0.116	0.561	0.535	0.997	0.697	0.590	0.851	0.241
34	multi-qa-mpnet-base-cos-v1	0.34	0.116	0.561	0.535	0.997	0.697	0.590	0.851	0.241
35	AIDA-UPM_MSTSb_paraphrase-multilingual-MiniLM-L12-v2	0.34	0.117	0.562	0.536	0.997	0.697	0.591	0.851	0.243
36	all-MiniLM-L12-v2	0.28	0.117	0.561	0.536	0.996	0.697	0.590	0.850	0.239
37	all-distilroberta-v1	0.28	0.117	0.562	0.536	0.997	0.697	0.591	0.851	0.243
38	all-roberta-large-v1	0.33	0.117	0.562	0.536	0.997	0.697	0.591	0.851	0.243
39	facebook-dpr-question_encoder-multiset-base	0.79	0.117	0.561	0.536	0.996	0.697	0.590	0.850	0.239
40	hroth_psais-paraphrase-multilingual-MiniLM-L12-v2-20shot	0.32	0.117	0.561	0.536	0.996	0.697	0.590	0.850	0.239
41	ibaucells_paraphrase-multilingual-mpnet-base-v2_tecla_label2_8	0.24	0.118	0.562	0.536	0.996	0.697	0.590	0.850	0.240
42	paraphrase-MiniLM-L12-v2	0.38	0.120	0.563	0.536	0.996	0.697	0.591	0.850	0.242
43	msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned	0.41	0.121	0.563	0.536	0.995	0.697	0.591	0.850	0.240
44	msmarco-roberta-base-v2	0.42	0.121	0.563	0.536	0.995	0.697	0.591	0.850	0.240
45	all-MiniLM-L6-v2	0.42	0.122	0.563	0.536	0.994	0.697	0.591	0.849	0.237
46	distilbert-base-nli-mean-tokens	0.67	0.122	0.563	0.537	0.995	0.697	0.591	0.850	0.241
47	distilbert-base-nli-max-tokens	0.82	0.125	0.563	0.537	0.992	0.697	0.591	0.848	0.237
48	quora-distilbert-base	0.49	0.125	0.564	0.537	0.994	0.697	0.591	0.849	0.241
49	paraphrase-mpnet-base-v2	0.49	0.126	0.564	0.537	0.992	0.697	0.591	0.849	0.238
50	distilbert-multilingual-nli-stsb-quora-ranking	0.88	0.127	0.564	0.537	0.991	0.697	0.591	0.848	0.236
51	quora-distilbert-multilingual	0.88	0.127	0.564	0.537	0.991	0.697	0.591	0.848	0.236
52	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_30_Epochs	0.56	0.130	0.565	0.538	0.990	0.697	0.592	0.847	0.236
53	jfarray_Model_paraphrase-multilingual-mpnet-base-v2_10_Epochs	0.59	0.135	0.566	0.538	0.987	0.697	0.592	0.846	0.235

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
54	bert-base-nli-cls-token	0.55	0.106	0.558	0.533	1.000	0.696	0.588	0.851	0.237
55	bert-large-nli-mean-tokens	0.57	0.106	0.558	0.533	1.000	0.696	0.588	0.851	0.237
56	nli-bert-base-cls-pooling	0.55	0.106	0.558	0.533	1.000	0.696	0.588	0.851	0.237
57	nli-bert-large	0.57	0.106	0.558	0.533	1.000	0.696	0.588	0.851	0.237
58	bert-large-nli-max-tokens	0.69	0.108	0.559	0.534	1.000	0.696	0.589	0.851	0.240
59	msmarco-distilbert-base-dot-prod-v3	0.56	0.108	0.559	0.534	1.000	0.696	0.589	0.851	0.240
60	nli-bert-large-max-pooling	0.69	0.108	0.559	0.534	1.000	0.696	0.589	0.851	0.240
61	bert-base-nli-mean-tokens	0.62	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
62	bert-base-nli-wkpooling	0.62	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
63	bert-large-nli-cls-token	0.65	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
64	bert-large-nli-stsb-mean-tokens	0.43	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
65	cointegrated_rubert-base-cased-dp-paraphrase-detection	0.76	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
66	distilbert-base-nli-stsb-quora-ranking	0.45	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
67	distiluse-base-multilingual-cased-v2	0.33	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
68	distiluse-base-multilingual-cased	0.33	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
69	msmarco-roberta-base-ance-firstp	0.99	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
70	msmarco-roberta-base-ance-fristp	0.99	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
71	multi-qa-MiniLM-L6-dot-v1	0.59	0.109	0.559	0.534	0.999	0.696	0.589	0.851	0.238
72	nli-bert-base	0.62	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
73	nli-bert-large-cls-pooling	0.65	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
74	nli-distilbert-base-max-pooling	0.50	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
75	nli-distilbert-base	0.59	0.109	0.560	0.534	1.000	0.696	0.589	0.852	0.242
76	bert-base-nli-max-tokens	0.67	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
77	distilroberta-base-msmarco-v1	0.32	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
78	distilroberta-base-msmarco-v2	0.28	0.111	0.560	0.534	0.999	0.696	0.589	0.851	0.239
79	msmarco-MiniLM-L-6-v3	0.27	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
80	msmarco-distilbert-dot-v5	0.72	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235
81	msmarco-distilroberta-base-v2	0.28	0.111	0.560	0.534	0.999	0.696	0.589	0.851	0.239
82	nli-bert-base-max-pooling	0.67	0.111	0.559	0.534	0.997	0.696	0.589	0.850	0.235

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
83	facebook-dpr-ctx_encoder-single-nq-base	0.66	0.113	0.560	0.534	0.996	0.696	0.589	0.849	0.234
84	msmarco-distilbert-base-v2	0.30	0.113	0.560	0.534	0.996	0.696	0.589	0.849	0.234
85	msmarco-distilbert-base-v3	0.30	0.113	0.560	0.535	0.997	0.696	0.589	0.850	0.238
86	multi-qa-MiniLM-L6-cos-v1	0.28	0.113	0.560	0.535	0.997	0.696	0.589	0.850	0.238
87	multi-qa-mpnet-base-dot-v1	0.53	0.116	0.560	0.535	0.995	0.696	0.589	0.849	0.234
88	JoBeer_paraphrase-multilingual-MiniLM-L12-v2-eclass	0.46	0.117	0.561	0.535	0.995	0.696	0.590	0.849	0.235
89	multi-qa-distilbert-cos-v1	0.43	0.117	0.561	0.535	0.995	0.696	0.590	0.849	0.235
90	facebook-dpr-ctx_encoder-multiset-base	0.73	0.118	0.561	0.535	0.994	0.696	0.590	0.848	0.233
91	multi-qa-distilbert-dot-v1	0.46	0.118	0.561	0.536	0.995	0.696	0.590	0.849	0.237
92	DataikuNLP_paraphrase-MiniLM-L6-v2	0.41	0.120	0.561	0.536	0.994	0.696	0.590	0.849	0.234
93	paraphrase-MiniLM-L6-v2	0.41	0.120	0.561	0.536	0.994	0.696	0.590	0.849	0.234
94	paraphrase-albert-base-v2	0.45	0.120	0.561	0.536	0.994	0.696	0.590	0.849	0.234
95	nli-mpnet-base-v2	0.63	0.121	0.561	0.536	0.992	0.696	0.590	0.848	0.232
96	Huffon_paraphrase-multilingual-mpnet-base-v2-512	0.57	0.122	0.561	0.536	0.991	0.696	0.590	0.847	0.230
97	all-MiniLM-L12-v1	0.41	0.122	0.562	0.536	0.992	0.696	0.590	0.848	0.234
98	paraphrase-MiniLM-L3-v2	0.38	0.124	0.563	0.536	0.992	0.696	0.591	0.848	0.235
99	Hoax0930_paraphrase-multilingual-MiniLM-L12-v2	0.57	0.126	0.563	0.537	0.990	0.696	0.591	0.847	0.231
100	AIDA-UPM_MSTSb_paraphrase-xlm-r-multilingual-v1	0.46	0.129	0.564	0.537	0.990	0.696	0.591	0.847	0.234
101	DataikuNLP_paraphrase-albert-small-v2	0.42	0.129	0.564	0.537	0.990	0.696	0.591	0.847	0.234
102	distilroberta-base-paraphrase-v1	0.44	0.129	0.563	0.537	0.989	0.696	0.591	0.846	0.231
103	paraphrase-TinyBERT-L6-v2	0.48	0.129	0.563	0.537	0.989	0.696	0.591	0.846	0.231
104	paraphrase-albert-small-v2	0.42	0.129	0.564	0.537	0.990	0.696	0.591	0.847	0.234
105	paraphrase-distilroberta-base-v1	0.44	0.129	0.563	0.537	0.989	0.696	0.591	0.846	0.231
106	distiluse-base-multilingual-cased-v1	0.45	0.131	0.564	0.537	0.987	0.696	0.591	0.846	0.231
107	JoBeer_paraphrase-MiniLM-L6-v2-eclass	0.38	0.104	0.556	0.533	0.999	0.695	0.587	0.850	0.232
108	average_word_embeddings_komninos	0.71	0.104	0.556	0.533	0.999	0.695	0.587	0.850	0.232
109	gart-labor_paraphrase-MiniLM-L6-v2-eclass	0.38	0.104	0.556	0.533	0.999	0.695	0.587	0.850	0.232
110	msmarco-distilbert-base-tas-b	0.73	0.107	0.558	0.533	0.999	0.695	0.588	0.850	0.235
111	facebook-dpr-question_encoder-single-nq-base	0.81	0.109	0.558	0.533	0.996	0.695	0.588	0.849	0.230

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
112	msmarco-MiniLM-L6-cos-v5	0.26	0.109	0.558	0.534	0.997	0.695	0.588	0.850	0.234
113	LaBSE	0.49	0.113	0.559	0.534	0.995	0.695	0.589	0.849	0.230
114	allenai-specter	0.77	0.113	0.558	0.534	0.994	0.695	0.588	0.848	0.227
115	JasperYOU_paraphrase-multilingual-mpnet-base-v2-exp	0.45	0.115	0.560	0.535	0.995	0.695	0.589	0.849	0.232
116	msmarco-distilbert-multilingual-en-de-v2-tmp-trained-scratch	0.28	0.115	0.559	0.534	0.994	0.695	0.589	0.848	0.228
117	msmarco-roberta-base-v3	0.32	0.116	0.560	0.535	0.994	0.695	0.589	0.848	0.230
118	msmarco-distilbert-base-v4	0.39	0.118	0.560	0.535	0.991	0.695	0.589	0.847	0.225
119	msmarco-distilbert-cos-v5	0.36	0.118	0.560	0.535	0.992	0.695	0.589	0.847	0.229
120	gemasphi_setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.30	0.124	0.561	0.536	0.990	0.695	0.590	0.846	0.228
121	new5558_chula-course-paraphrase-multilingual-mpnet-base-v2	0.40	0.127	0.562	0.536	0.987	0.695	0.590	0.845	0.226
122	clip-ViT-B-32-multilingual-v1	0.82	0.098	0.554	0.531	1.000	0.694	0.586	0.850	0.228
123	bert-base-nli-stsb-wkpooling	0.39	0.099	0.554	0.531	1.000	0.694	0.586	0.850	0.229
124	average_word_embeddings_levy_dependency	0.77	0.102	0.555	0.532	0.999	0.694	0.587	0.850	0.228
125	average_word_embeddings_glove.6B.300d	0.45	0.103	0.556	0.532	0.999	0.694	0.587	0.850	0.230
126	average_word_embeddings_glove.840B.300d	0.55	0.108	0.557	0.533	0.996	0.694	0.588	0.849	0.228
127	Hoax0930_tf_paraphrase-multilingual-MiniLM-L12-v2	0.15	0.109	0.557	0.533	0.995	0.694	0.588	0.848	0.226
128	msmarco-MiniLM-L-12-v3	0.23	0.109	0.557	0.533	0.995	0.694	0.588	0.848	0.226
129	Hoax0930_paraphrase-multilingual-mpnet-base-v2	0.52	0.115	0.558	0.534	0.992	0.694	0.588	0.847	0.224
130	hroth01_psais-paraphrase-multilingual-MiniLM-L12-v2-50shot	0.37	0.115	0.558	0.534	0.992	0.694	0.588	0.847	0.224
131	msmarco-bert-base-dot-v5	0.89	0.117	0.558	0.534	0.990	0.694	0.588	0.846	0.220
132	gemasphi_real-setfit-ss-paraphrase-multilingual-mpnet-base-v2	0.30	0.113	0.557	0.533	0.991	0.693	0.587	0.846	0.219
133	msmarco-MiniLM-L12-cos-v5	0.32	0.117	0.557	0.533	0.987	0.693	0.587	0.844	0.213
134	nq-distilbert-base-v1	0.36	0.118	0.558	0.534	0.989	0.693	0.588	0.845	0.218
135	Prompsit_paraphrase-bert-en	0.64	0.091	0.549	0.528	0.996	0.691	0.583	0.846	0.207
136	bert-base-wikipedia-sections-mean-tokens	0.99	0.108	0.553	0.531	0.989	0.691	0.585	0.843	0.205
137	aditeyabaral-bert-base-cased	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
138	aditeyabaral-contrastive-roberta-base	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
139	aditeyabaral-distilbert-base-cased	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
140	aditeyabaral-roberta-base	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200

#	Model	θ	S	A	P	R	F1	F.5	F2	MCC
141	aditeyabaral-xlm-roberta-base	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
142	moshew_paraphrase-mpnet-base-v2_SetFit_emotions	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
143	orenpereg_paraphrase-mpnet-base-v2_sst2_64samps	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
144	valurank_paraphrase-mpnet-base-v2-offensive	0.00	0.076	0.543	0.525	1.000	0.689	0.580	0.847	0.200
145	moshew_paraphrase-mpnet-base-v2_SetFit_sst2	0.00	0.080	0.543	0.525	0.996	0.688	0.580	0.845	0.191
146	Hoax0930_tf_paraphrase-multilingual-mpnet-base-v2	0.11	0.102	0.548	0.528	0.985	0.688	0.582	0.840	0.185

Dodatak: Vrijednosti F1-mjere s obzirom na dimenzije vektorskih reprezentacija

Rezultati predstavljeni u ovom dodatku rezultat su eksperimenata kojima je za cilj bio utvrditi povezanost broja dimenzija jezičnih modela temeljenih na dubokom učenju s njihovim učinkom otkrivanja sličnosti mjerenima F1-mjerom. Za svaku od četiri krivulje prikazane na slikama korišteno je 256 uređenih parova vrijednosti (broj dimenzija, vrijednost F1-mjere).

Tablica 39. F1-mjera kao funkcija dimenzija VP

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
16	0.809	0.606	0.797	0.598
32	0.808	0.680	0.794	0.662
48	0.810	0.660	0.798	0.644
64	0.811	0.675	0.795	0.692
80	0.809	0.678	0.802	0.657
96	0.808	0.713	0.799	0.674
112	0.813	0.705	0.807	0.700
128	0.814	0.735	0.810	0.723
144	0.812	0.753	0.811	0.732
160	0.812	0.735	0.801	0.725
176	0.810	0.728	0.799	0.730
192	0.812	0.721	0.813	0.720
208	0.812	0.721	0.811	0.680
224	0.812	0.730	0.799	0.741
240	0.811	0.715	0.803	0.727
256	0.815	0.718	0.808	0.719
272	0.813	0.733	0.812	0.694
288	0.810	0.708	0.803	0.703
304	0.813	0.744	0.811	0.747
320	0.813	0.737	0.810	0.741
336	0.812	0.726	0.805	0.716
352	0.812	0.736	0.807	0.737
368	0.814	0.736	0.809	0.722
384	0.812	0.742	0.800	0.727
400	0.813	0.729	0.805	0.712
416	0.816	0.724	0.807	0.703
432	0.813	0.736	0.804	0.731
448	0.813	0.722	0.812	0.721
464	0.810	0.741	0.802	0.729
480	0.813	0.743	0.816	0.704

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
496	0.813	0.719	0.805	0.738
512	0.813	0.721	0.814	0.731
528	0.811	0.705	0.810	0.733
544	0.815	0.739	0.806	0.410
560	0.814	0.724	0.808	0.723
576	0.814	0.713	0.813	0.706
592	0.815	0.707	0.809	0.721
608	0.814	0.728	0.813	0.739
624	0.814	0.745	0.812	0.719
640	0.814	0.729	0.807	0.737
656	0.813	0.720	0.803	0.721
672	0.814	0.750	0.807	0.720
688	0.814	0.707	0.807	0.712
704	0.814	0.718	0.805	0.696
720	0.814	0.732	0.812	0.731
736	0.812	0.735	0.802	0.739
752	0.814	0.727	0.806	0.703
768	0.814	0.732	0.806	0.729
784	0.812	0.727	0.811	0.735
800	0.811	0.718	0.803	0.712
816	0.815	0.719	0.805	0.722
832	0.814	0.722	0.809	0.730
848	0.814	0.727	0.813	0.718
864	0.815	0.737	0.806	0.730
880	0.813	0.713	0.815	0.718
896	0.812	0.727	0.810	0.740
912	0.812	0.736	0.800	0.715
928	0.813	0.733	0.809	0.717
944	0.813	0.719	0.802	0.667
960	0.814	0.744	0.808	0.743
976	0.813	0.730	0.809	0.724
992	0.814	0.752	0.802	0.727
1008	0.814	0.732	0.807	0.731
1024	0.811	0.737	0.806	0.709
1040	0.814	0.725	0.800	0.727
1056	0.812	0.735	0.804	0.720
1072	0.814	0.716	0.811	0.726
1088	0.814	0.743	0.810	0.732
1104	0.815	0.740	0.815	0.725
1120	0.814	0.720	0.809	0.719
1136	0.813	0.732	0.808	0.734
1152	0.813	0.731	0.806	0.716
1168	0.812	0.731	0.805	0.675
1184	0.814	0.726	0.810	0.727
1200	0.814	0.734	0.810	0.718
1216	0.813	0.726	0.812	0.722

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
1232	0.813	0.740	0.806	0.701
1248	0.805	0.712	0.803	0.719
1264	0.809	0.737	0.805	0.716
1280	0.801	0.720	0.801	0.679
1296	0.810	0.728	0.812	0.717
1312	0.800	0.726	0.801	0.736
1328	0.810	0.713	0.811	0.720
1344	0.803	0.738	0.804	0.721
1360	0.803	0.722	0.803	0.721
1376	0.803	0.727	0.802	0.733
1392	0.807	0.732	0.809	0.739
1408	0.810	0.731	0.810	0.717
1424	0.816	0.713	0.812	0.716
1440	0.806	0.733	0.804	0.711
1456	0.806	0.746	0.806	0.749
1472	0.808	0.720	0.810	0.725
1488	0.802	0.723	0.802	0.704
1504	0.816	0.744	0.808	0.707
1520	0.811	0.740	0.811	0.740
1536	0.807	0.726	0.814	0.725
1552	0.816	0.722	0.810	0.717
1568	0.807	0.724	0.815	0.738
1584	0.806	0.740	0.806	0.721
1600	0.805	0.726	0.805	0.721
1616	0.806	0.727	0.808	0.675
1632	0.811	0.742	0.810	0.697
1648	0.808	0.724	0.808	0.721
1664	0.801	0.726	0.801	0.712
1680	0.803	0.720	0.807	0.723
1696	0.805	0.718	0.806	0.719
1712	0.813	0.733	0.813	0.722
1728	0.811	0.719	0.811	0.724
1744	0.816	0.724	0.815	0.720
1760	0.810	0.739	0.807	0.723
1776	0.814	0.727	0.805	0.716
1792	0.806	0.731	0.808	0.734
1808	0.804	0.722	0.805	0.727
1824	0.812	0.735	0.815	0.717
1840	0.811	0.719	0.803	0.713
1856	0.815	0.725	0.808	0.727
1872	0.803	0.727	0.806	0.734
1888	0.805	0.724	0.807	0.675
1904	0.805	0.730	0.817	0.739
1920	0.812	0.730	0.812	0.713
1936	0.808	0.733	0.808	0.736
1952	0.815	0.721	0.814	0.717

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
1968	0.805	0.726	0.805	0.724
1984	0.808	0.716	0.814	0.728
2000	0.805	0.730	0.806	0.721
2016	0.804	0.744	0.807	0.720
2032	0.809	0.739	0.809	0.735
2048	0.817	0.724	0.810	0.719
2064	0.808	0.730	0.808	0.648
2080	0.809	0.738	0.811	0.720
2096	0.803	0.728	0.807	0.739
2112	0.803	0.734	0.803	0.739
2128	0.812	0.729	0.816	0.732
2144	0.804	0.737	0.804	0.722
2160	0.818	0.731	0.814	0.730
2176	0.809	0.737	0.811	0.721
2192	0.804	0.740	0.810	0.740
2208	0.810	0.734	0.809	0.748
2224	0.815	0.722	0.815	0.745
2240	0.810	0.731	0.810	0.741
2256	0.817	0.738	0.807	0.752
2272	0.808	0.731	0.810	0.730
2288	0.804	0.715	0.807	0.720
2304	0.804	0.726	0.810	0.717
2320	0.805	0.744	0.808	0.726
2336	0.801	0.725	0.802	0.724
2352	0.815	0.738	0.815	0.724
2368	0.808	0.721	0.808	0.721
2384	0.815	0.731	0.818	0.720
2400	0.815	0.722	0.816	0.722
2416	0.809	0.724	0.807	0.729
2432	0.809	0.730	0.809	0.725
2448	0.805	0.721	0.805	0.722
2464	0.803	0.736	0.804	0.732
2480	0.816	0.722	0.814	0.722
2496	0.804	0.732	0.808	0.711
2512	0.817	0.722	0.816	0.731
2528	0.805	0.735	0.805	0.724
2544	0.806	0.727	0.808	0.729
2560	0.808	0.742	0.808	0.729
2576	0.801	0.726	0.802	0.725
2592	0.802	0.735	0.807	0.732
2608	0.813	0.734	0.813	0.721
2624	0.817	0.732	0.816	0.714
2640	0.806	0.745	0.808	0.730
2656	0.805	0.731	0.812	0.717
2672	0.801	0.737	0.801	0.722
2688	0.808	0.739	0.809	0.659

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
2704	0.806	0.744	0.806	0.720
2720	0.805	0.744	0.806	0.648
2736	0.808	0.726	0.809	0.713
2752	0.807	0.726	0.806	0.711
2768	0.806	0.731	0.806	0.718
2784	0.811	0.728	0.806	0.712
2800	0.808	0.738	0.808	0.719
2816	0.805	0.736	0.805	0.738
2832	0.810	0.746	0.805	0.733
2848	0.803	0.730	0.803	0.725
2864	0.806	0.732	0.806	0.725
2880	0.813	0.743	0.814	0.729
2896	0.808	0.745	0.809	0.728
2912	0.818	0.722	0.801	0.699
2928	0.804	0.733	0.806	0.710
2944	0.812	0.739	0.802	0.721
2960	0.802	0.741	0.808	0.739
2976	0.807	0.739	0.809	0.735
2992	0.808	0.735	0.807	0.731
3008	0.815	0.730	0.806	0.728
3024	0.814	0.719	0.806	0.728
3040	0.814	0.744	0.814	0.727
3056	0.812	0.726	0.812	0.727
3072	0.813	0.724	0.810	0.727
3088	0.802	0.731	0.802	0.732
3104	0.805	0.741	0.815	0.719
3120	0.801	0.742	0.801	0.735
3136	0.809	0.734	0.808	0.680
3152	0.808	0.733	0.808	0.729
3168	0.799	0.721	0.807	0.648
3184	0.807	0.734	0.808	0.730
3200	0.803	0.751	0.808	0.729
3216	0.806	0.723	0.817	0.727
3232	0.804	0.738	0.807	0.722
3248	0.807	0.727	0.808	0.711
3264	0.818	0.724	0.818	0.717
3280	0.805	0.717	0.809	0.725
3296	0.808	0.727	0.815	0.714
3312	0.807	0.724	0.809	0.734
3328	0.807	0.727	0.806	0.722
3344	0.819	0.738	0.814	0.719
3360	0.812	0.736	0.802	0.728
3376	0.806	0.732	0.809	0.721
3392	0.816	0.732	0.803	0.689
3408	0.805	0.727	0.805	0.718
3424	0.801	0.728	0.804	0.727

Broj dimenzija	F1 (MSRP, Doc2Vec)[1]	F1 (P4PIN, Doc2Vec)[2]	F1 (MSRP, Word2Vec)[3]	F1 (P4PIN, Word2Vec)[4]
3440	0.807	0.732	0.803	0.730
3456	0.808	0.724	0.806	0.728
3472	0.808	0.732	0.807	0.721
3488	0.816	0.722	0.817	0.711
3504	0.805	0.724	0.806	0.732
3520	0.811	0.721	0.812	0.717
3536	0.805	0.714	0.815	0.716
3552	0.809	0.731	0.809	0.615
3568	0.807	0.733	0.803	0.722
3584	0.808	0.727	0.808	0.721
3600	0.812	0.734	0.813	0.740
3616	0.806	0.726	0.806	0.737
3632	0.803	0.734	0.816	0.742
3648	0.810	0.729	0.816	0.736
3664	0.807	0.730	0.808	0.734
3680	0.808	0.735	0.803	0.729
3696	0.805	0.733	0.806	0.716
3712	0.807	0.728	0.806	0.711
3728	0.816	0.722	0.816	0.732
3744	0.808	0.724	0.810	0.706
3760	0.810	0.734	0.807	0.643
3776	0.809	0.740	0.806	0.736
3792	0.802	0.726	0.803	0.740
3808	0.817	0.736	0.818	0.725
3824	0.809	0.737	0.815	0.731
3840	0.810	0.729	0.803	0.729
3856	0.810	0.735	0.817	0.722
3872	0.808	0.729	0.808	0.718
3888	0.809	0.740	0.807	0.741
3904	0.815	0.745	0.808	0.725
3920	0.807	0.731	0.801	0.706
3936	0.807	0.724	0.804	0.727
3952	0.812	0.724	0.810	0.721
3968	0.805	0.727	0.805	0.738
3984	0.816	0.741	0.810	0.741
4000	0.808	0.736	0.810	0.728
4016	0.806	0.729	0.808	0.720
4032	0.810	0.733	0.816	0.717
4048	0.803	0.737	0.806	0.719
4064	0.809	0.738	0.809	0.732
4080	0.809	0.736	0.808	0.731
4096	0.815	0.727	0.816	0.733

[1] F1-mjera MSRP (Doc2Vec DBoW)
[2] F1-mjera P4PIN (Doc2Vec DBoW)
[3] F1-mjera MSRP (Word2Vec CBOW)
[4] F1-mjera P4PIN (Word2Vec CBOW)

Dodatak: Primjeri krivih oznaka korpusa MSRP

U ovom dodatku naveden je samo **mali broj primjera** inače mnogobrojnih dvojbenih oznaka parafraziranih parova[1] u korpusu MSRP.

Tablica 40. Primjeri burzovnih izvješća korpusa MSRP

Oznaka	Tekst1	Tekst2
0	The Dow Jones industrial average fell 98.30, or 1.1 percent, while bond values fell, too.	The Dow Jones industrial average finished the day down 98.32 points at 9,011.53.
1	The broad Standard & Poor's 500 Index <.SPX> lost 6 points, or 0.71 percent, to 927.	The broad Standard & Poors 500-stock index was down 4.77 points to 929.62.
1	Axcan's shares closed down 63 Canadian cents, or 4 percent, at C$16.93 in Toronto on Tuesday.	Axcan's shares were down 3.8 percent, or 66 Canadian cents, at C$16.90 in Toronto on Tuesday.
0	The broader Standard & Poor's 500 Index gained 16.02 points, or 1.62 percent, at 1,004.63.	The technology-laced Nasdaq Composite Index added 28.73 points, or 1.77 percent, at 1,655.22.
0	The tech-heavy Nasdaq Stock Markets composite index added 1.16 points to 1,504.04.	The Nasdaq Composite index, full of technology stocks, was lately up around 18 points.
0	The broader Standard & Poor's 500 Index .SPX eased 7.57 points, or 0.76 percent, at 990.94.	The technology-laced Nasdaq Composite Index was down 25.36 points, or 1.53 percent, at 1,628.26.

Tablica 41. Primjeri izvješća promjene tečajeva valuta korpusa MSRP

Oznaka	Tekst1	Tekst2
1	The Swiss franc rose nearly a third of a centime against the dollar and was last at 1.2998 <CHF=> to the greenback.	The Swiss franc rose three quarters of a percent against the dollar and was last at 1.2980 to the greenback.
0	Sterling was down 0.8 percent against the dollar at $1.5875 GBP= .	The dollar rose 0.15 percent against the Japanese currency to 115.97 yen.
0	The reports helped overcome investor jitters after the euro briefly hit an all-time high against the dollar Tuesday.	Stocks slipped at the open after the euro hit record highs against the dollar.
0	Around 9:00 a.m. EDT (1300 GMT), the euro was at $1.1566 against the dollar, up 0.07 percent on the day.	Against the Swiss franc the dollar was at 1.3289 francs, up 0.5 percent on the day.
0	The dollar was last at $1.1149 to the euro, close to its strongest level since April 30.	The dollar pushed as high as $1.1115 to the euro in early trade, extending Tuesday's one percent rally to hit its strongest level since April 30.

Tablica 42. Primjeri najava iz svijeta filma korpusa MSRP

Oznaka	Tekst1	Tekst2
1	His other films include "Malcolm X," "Summer of Sam" and "Jungle Fever."	His movies include "Malcolm X," "Summer of Sam," "Jungle Fever" and "Do the Right Thing."
0	The street-racing sequel "2 Fast 2 Furious" won	The PG-13 sequel "2 Fast 2 Furious" raked in an

1 Zbog uštede prostora ovdje je naveden samo mali broj „slučajeva”, jer se na 500-njak takvih otkrivenih dvojbenih primjera, odustalo se od pokušaja popravljanja MSRP korpusa. Datoteka s primjerima dostupna je na https://vrbanec.com/corpora.

Oznaka	Tekst1	Tekst2
	the pole position at the box office, taking in an estimated $52.1 million in its opening weekend.	estimated $52.1 million during its opening weekend, jumping over last weekend's catch, "Finding Nemo."
0	Shattered Glass,"starring Hayden Christensen as Stephen Glass, debuted well with $80,000 in eight theaters.	"Shattered Glass" _ starring Hayden Christensen as Stephen Glass, The New Republic journalist fired for fabricating stories _ debuted well with $80,000 in eight theaters.
0	Directed by Jonathan Lynn from a script by Elizabeth Hunter and Saladin K. Patterson.	Whole Nine Yards," and written by first-time screenwriters Elizabeth Hunter and Saladin K. Patterson.

Tablica 43. Primjeri očito krivih oznaka korpusa MSRP

Oznaka	Tekst1	Tekst2
0	"It's just too important to keeping crime down," he said of Operation Impact, which began Jan. 3.	"It's just too important to keeping crime down in the city to let it lapse," the mayor said of Operation Impact.
0	Germany's Foreign Ministry said it believed the passengers were from the northern states of Lower Saxony and Schleswig-Holstein, but had no further details.	Germany said most of the passengers were from the northern states of Lower Saxony and Schleswig-Holstein.
0	Proving that the Millville son's sacrifice would not go unsung, Mayor James Quinn ordered all city flags flown at half-mast for the next 30 days.	In Millville yesterday, Mayor James Quinn ordered all city flags flown at half-staff for the next 30 days.
0	Pratt &Whitney had said that 75 per cent of the engine equipment would be outsourced to Europe, with final assembly in Germany.	Pratt & Whitney had said that if it won the contract 75 per cent of the engine equipment would be outsourced to European suppliers, with final assembly in Germany.
0	Now, Blanca's American husband, 63-year-old Roger Lawrence Strunk, faces a murder indictment issued in February by the Philippine government, which says he's the leading suspect.	Now, Blanca's husband, 63-year-old Roger Lawrence Strunk of Tracy, faces a murder indictment issued by the Philippine government in February.
0	"Certainly what we know suggests that we should take what they're saying very seriously," he added.	"We don't know everything [but] what we know suggests that we should take what they're saying seriously," he said.
0	On Wall Street, trading resumed with some glitches from the blackout that continued to affect parts of New York City.	Stocks barely budged Friday as trading resumed with some glitches from the blackout in New York.
0	Danbury prosecutor Warren Murray could not be reached for comment Monday.	Prosecutors could not be reached for comment after the legal papers were obtained late Monday afternoon.
0	Both bidders agreed to assume about $90 million in debt owed on the planes.	Wexford had agreed to assume about $90 million in debt to buy the planes and certificate.
0	In 1999, the building's owners, the Port Authority of New York and New Jersey, issued guidelines to upgrade the fireproofing to a thickness of 1{ inches.	The NIST discovered that in 1999 the Port Authority issued guidelines to upgrade the fireproofing to a thickness of 1 1/2 inches.
0	Palm plans to issue about 13.9 million shares of Palm common stock to Handspring's shareholders, on a fully diluted basis.	Palm will issue approximately 13.9 million shares of Palm common stock to Handspring's shareholders.
0	Defense Secretary Donald Rumsfeld is awaiting recommendations from his commanders.	Rumsfeld is awaiting recommendations from his commanders about troop needs in Iraq.
0	With the test, Hubbard said, "I believe that we have found the smoking gun.	"We have found the smoking gun," investigating board member Scott Hubbard said.
0	The Pentagon, saying that Boykin requested it, is investigating his remarks.	The Pentagon has begun an official investigation into Boykin's remarks.

Oznaka	Tekst1	Tekst2
0	But for more than a century, an untold amount of money intended for some of the nation's poorest residents was lost, stolen or never collected.	For more than a century, an undetermined amount of money was lost, stolen, or never collected.
0	Sony Ericsson also said it would shut down its GSM/UMTS R&D center in Munich, Germany, to increase profitability.	Sony Ericsson also said it plans to close its R&D site in Munich, Germany, for GSM and UMTS handsets.
0	Though that slower spending made 2003 look better, many of the expenditures actually will occur in 2004.	Though that slower spending made 2003 look better, many of the expenditures will actually occur in 2004, making that year's shortfall worse.

Tablica 44. Primjeri poslovnih izvještaja korpusa MSRP

Oznaka	Tekst1	Tekst2
1	Second quarter sales came in at $645 million, up from $600 million the year before, AMD said.	Revenue in the second quarter ended June 29 was $645 million, up from $600 million a year ago.
0	In the second quarter, Anadarko now expects volume of 46 million BOE, down from 48 million BOE.	Production for the second quarter was cut to 46 million barrels from 48 million barrels.
1	Dell has about 32 percent of the U.S. market, but much lower share in the rest of the world.	Dell has 32 percent of the PC market in the United States, but it has only a 10 percent share in the rest of the world.
0	During the same quarter last year, EDS declared a profit of $354 million, or 72 cents per share.	EDS reported a first-quarter loss of $126 million, or 26 cents per share.
0	Excluding one-time items, the company enjoyed a profit of 6 cents a share.	Excluding one-time items, it expects profit of 11 cents to 15 cents a share.
1	General Motors Corp. posted a record 8.4 percent improvement in 2000.	General Motors Corp. posted the best-ever improvement in 2000 at 8.4 percent.
0	The 30-year bond US30YT=RR firmed 26/32, taking its yield to 4.17 percent, after hitting another record low of 4.16 percent.	The 30-year bond US30YT=RR firmed 14/32, taking its yield to 4.19 percent from 4.22 percent.
0	China accounted for about 14 percent of Motorola's sales last year, and the company has large manufacturing operations there.	China accounted for 14% of Motorola's $26.7 billion in sales last year.
1	If achieved it would represent an increase of 10 per cent from the same quarter a year ago.	That would represent an increase of some 10 per cent from the year before, it added.

Tablica 45. Zbunjujući primjeri korpusa MSRP

Oznaka	Tekst1	Tekst2
0	Preliminary reports were that the men were not seen together at the airport, sources said.	The men had Pakistani passports and reportedly were seen together at the airport earlier in the evening, law enforcement sources said.
0	Details of the research appear in the Nov. 5 issue of the Journal of the American Medical Association.	The results, published in the Journal of the American Medical Association, involved just 47 heart attack patients.
1	The results will be published the July 10 issue of the journal Nature.	The results appear in Thursday’s issue of the journal Nature.
0	Tidmarsh will compete in today's third round.	Two kids from Michigan are in today's third round.
0	One, Capt. Doug McDonald, remained hospitalized in critical condition on Thursday.	Her 20-year-old sister, Allyson, was severely burned and remained hospitalized in critical condition.
1	Five Big East schools, Connecticut, Pittsburgh, Rutgers, Virginia Tech and West Virginia, filed the lawsuit June 6.	Pittsburgh, West Virginia, Rutgers, Connecticut and Virginia Tech filed the lawsuit this month in Hartford, Conn.

Oznaka	Tekst1	Tekst2
1	That investigation closed without any charges being laid.	The investigation was closed without charges in 2001.
1	But Senate criticism of the House plan Tuesday came on several fronts.	And several Senate Republicans are cranky about the House map.
1	Jason Giambi capitalized with an RBI single to center.	Jason Giambi contributed a two-out RBI single.
0	Ernst & Young has denied any wrongdoing and plans to fight the allegations.	Ernst & Young has denied the SEC's claims, and called its recommendations "irresponsible".

Tablica 46. Primjeri činjenica koje se ne mogu drugačije izraziti korpusa MSRP

Oznaka	Tekst1	Tekst2
0	The legislation came after U.S. District Judge Lee R. West in Oklahoma City ruled last week that the FTC lacked authority to run the registry.	U.S. District Judge Lee R. West ruled Tuesday in Oklahoma City that the FTC lacks authority to run the registry.
1	Legato stockholders will get 0.9 of a share of EMC stock for every share of Legato they own.	Under terms of the agreement, Legato stockholders will receive 0.9 shares of EMC common stock for each Legato share they hold.
1	A base configuration with a 2.0GHz Intel Celeron processor, 128M bytes of memory, a 40G-byte hard drive, and a CD-ROM drive costs US$729.	A base configuration with a 2.4GHz Pentium 4, 128MB of RAM, a 40GB hard drive, and a CD-ROM drive costs $699.
1	Ruffner, 45, doesn't yet have an attorney in the murder charge, authorities said.	Ruffner, 45, does not have a lawyer on the murder charge, authorities said.