УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Јелена Б. Граовац

ПРИЛОГ МЕТОДАМА КЛАСИФИКАЦИЈЕ ТЕКСТА: МАТЕМАТИЧКИ МОДЕЛИ И ПРИМЕНЕ

докторска дисертација

Београд, 2014.

UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Jelena B. Graovac

PRILOG METODAMA KLASIFIKACIJE TEKSTA: MATEMATIČKI MODELI I PRIMENE

doktorska disertacija

Beograd, 2014.

UNIVERSITY OF BELGRADE
FACULTY OF MATHEMATICS

Jelena B. Graovac

CONTRIBUTION TO TEXT CATEGORIZATION METHODS: MATHEMATICAL MODELS AND APPLICATIONS

Doctoral dissertation

Belgrade, 2014.

Mentor:
prof. dr Gordana Pavlović-Lažetić, redovni profesor,
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:
dr Duško Vitas, vanredni profesor,
Univerzitet u Beogradu, Matematički fakultet
prof. dr Ivan Obradović, redovni profesor,
Univerzitet u Beogradu, Rudarsko-geološki fakultet

Datum odbrane: _____

Posvećeno mom prerano preminulom ocu,
Branislavu Tomaševiću (1945-1980)

Naslov disertacije: Prilog metodama klasifikacije teksta: matematički modeli i primene

Rezime:

U svetu u kome živimo, internet i digitalni zapis učinili su da ogromne količine sirovih podataka postanu dostupne širokoj javnosti. Jedan američki menadžer je još davno izjavio: "Računari su nam obećali fontanu mudrosti, a ovo što smo dobili je poplava podataka" [20]. Sirovi podaci, neadekvatno strukturirani i različitih formata, sadržaja i kvaliteta su retko od koristi. Neophodno ih je pripremiti, analizirati i na osnovu toga doći do informacija i znanja koja na taj način stiču neprocenjivu vrednost.

Istraživanje podataka (eng. data mining) je interdisciplinarno polje informatike koje se bavi automatskim ili polu-automatskim otkrivanjem znanja u podacima. Njegov osnovni zadatak je netrivijalna ekstrakcija informacija iz podataka, i to informacija koje su implicitne, prethodno nepoznate i potencijalno korisne. Koriste se metode koje su u preseku veštačke inteligencije, mašinskog učenja, statistike i sistema baza podataka [97]. Zadaci koji se rešavaju u okviru *Istraživanja podataka* mogu biti prediktivni (klasifikacija, regresija, analiza vremenskih serija) ili deskriptivni (klasterovanje, sumarizacija, pravila pridruživanja, analiza redosleda, otkrivanje anomalija).

U okviru ove doktorske disertacije bavimo se problemom klasifikacije tekstova na osnovu njihovog sadržaja. Smatra se da je preko 80% dostupnih informacija sačuvano u tekstualnom obliku. Većina informacija je zapisana prirodnim jezikom, odnosno jezikom koji koriste ljudi za svakodnevnu komunikaciju. Za očekivati je da će tehnologije automatske obrade podataka zapisanih prirodnim jezikom postati vodeće u svetu. Glavni doprinos disertacije ogleda se u predstavljanju novih metoda za klasifikaciju tekstualnih dokumenata. Prva metoda predstavlja unapređenje metode razvijene u cilju otkrivanja autorstva teksta [38]. Metoda je zasnovana na predstavljanju dokumenta kao profila koji sadrži fiksiran broj n-grama bajtova koji se pojavljuju u dokumentu, i meri različitosti pomoću koje se određuje klasa kojoj dokument pripada. Ova metoda je jezički nezavisna i ne zahteva nikakvu prethodnu obradu teksta niti predznanje o sadržaju teksta ili jeziku na kome je tekst napisan. Druga metoda se zasniva na odabranim konceptima kao predstavnicima klasa koji se dobijaju iz srpskog wordnet-a, leksičko semantičke mreže za srpski jezik. Deo rezultata iz ove disertacije je sadržan u radovima [23, 27, 22, 21, 56, 26, 25, 24] koji su objavljeni, predati za objavljivanje ili su u fazi pripreme.

Disertacija je organizovana na sledeći način.

U glavi 1 je prikazan uvod u oblast klasifikacije podataka, u okviru koga su prikazane vrste klasifikacije, procena kvaliteta klasifikacije i primeri primene.

Poseban osvrt dat je na klasifikaciju tekstualnih dokumenata. Prikazani su različiti načini predstavljanja dokumenata kao jednog od najvažnijiih koraka u procesu klasifikacije. Predočeni su i mnogi problemi i izazovi koji se javljaju. Prikazani su korpusi klasifikovanih tekstova na srpskom, engleskom, kineskom i arapskom jeziku koji će biti korišćeni u daljem istraživanju. Uvodna glava završava se jednim filozofskim pogledom na proces klasifikacije.

Glava 2 daje pregled postojećih leksičkih resursa za srpski jezik [17] koji se razvijaju u okviru Grupe za jezičke tehnologije na Matematičkom fakultetu Univeziteta u Beogradu. Ideja je da se uključivanjem morfoloških, sintaksičkih i semantičkih informacija sadržanih u resursima unapredi proces klasifikacije tekstova na srpkom jeziku, kao jednom od morfološki bogatijih jezika. Predstavljeni su korpusi srpskog jezika, elektronski rečnik i srpski wordnet kao i raznovrsne tehnologije koje se koriste za njihovu obradu a koje se razvijaju u okviru Grupe.

U glavi 3 su prikazane postojeće metode mašinskog učenja koje su do sada imale veoma uspešnu primenu u procesu klasifikacije. Prikazane su metode zasnovane na drvetima odlučivanja, metode zasnovane na pravilima i rastojanju, statistički zasnovane metode, metode zasnovane na neuronskim mrežama i metode zasnovane na podržavajućim vektorima.

Nove metode za klasifikaciju teksta prikazane su u glavi 4. U okviru prve metode zasnovane na n-gramima bajtova, uvedeni su nova mera različitosti i novi težinski faktori u odnosu na osnovnu varijantu metode. Težinski faktori su dodeljeni n-gramima u okviru profila klasa, reflektujući značaj koji n-grami imaju za pripadajuću klasu. Smatra se da n-grami koji imaju veću frekvenciju a pripadaju manjem broju klasa imaju veći značaj za klasu kojoj pripadaju. Uvođenje ovih težinskih faktora rezultovalo je modifikacijom metode na dva načina: modifikacija na nivou mere različitosti i modifikacija na nivou profila klase. Druga metoda se odnosi na korišćenje informacija sadržanih u srpskom wordnetu i srpskom elektronskom rečniku u cilju klasifikacije teksta na srpskom jeziku. Ova metoda zasniva se na pridruživanju odabranih koncepata iz srpskog wordnet-a klasama, na osnovu kojih se izračunava mera pripadnosti klasi i vrši pridruživanje dokumenta nekoj od klasa.

Rezultati prikazanih novih metoda sumirani su u okviru glave 5. Na srpskom korpusu je prikazano poređenje prve metode i njenih modifikacija zasnovanih na n-gramima bajtova, karaktera i reči. Osnovna varijanta metode i njene modifikacije za n-grame bajtova, testirani su na korpusima na srpskom, engleskom, kineskom i arapskom jeziku, čime je demonstrirana jezička nezavisnost metode. U okviru Priloga 1 dodatno su predstavljeni svi rezultati dobijeni testiranjem metode za različite vrednosti parametara, za sve predstavljene mere različitosti, na svim pomenutim korpusima. Druga metoda testirana je samo na korpusu na srpskom jeziku.

Poređenje prikazanih rezultata sa drugim rezultataima iz ove oblasti dato je u glavi 6 a glava 7 prikazuje zaključke i pravce daljeg rada.

Ključne reči: klasifikacija teksta, obrada prirodnih jezika, n-grami, Wordnet

Naučna oblast: Računarstvo

Uža naučna oblast: Računarska obrada teksta

UDK broj: 004.832.2:025.4 (043.3)

Title of the dissertation: Contribution to text categorization methods: mathematical models and applications

Abstract:

We live in a world where the Internet and digital recording have made available huge amounts of raw data to the public. A frustrated management information systems executive a long time ago said: "Computers have promised us a fountain of wisdom but delivered a flood of data" [20]. Documents in their textual semi-structured data formats (or raw data), with different content and quality are rarely useful. It is necessary to prepare these raw data for analysis, to transform them into information and to transform information into invaluable knowledge.

Data mining, also known as knowledge-discovery in databases, is an interdisciplinary subfield of computer science which task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns. It can be defined as nontrivial extraction of implicit, previously unknown, and potentially useful information from data. The methods used are at the juncture of artificial intelligence, machine learning, statistics, database systems and business intelligence [97]. In general, data mining tasks can be classified into two categories: predictive (classification, regression, and times series) and descriptive (clustering, summarization, association rules, sequence analysis, anomaly detection).

This dissertation deals with the problem of automatic and semi-automatic content-based classification of natural language text documents. The main contribution of this thesis is development of new methods for text categorization. The first method is an improvement of Kešelj's method [38] to solving authorship attribution problem. The approach relies on a profile representation of restricted size of both document and a category, and a simple algorithm for comparing profiles. It is language independent and does not require any morphological analysis of texts, any preprocessing steps, or any prior information about document content or language. The second method is based on well-chosen concepts from lexical-semantic network Serbian wordnet, assigned to the corresponding categories. Parts from this dissertation have been described in papers [23, 27, 22, 21, 56, 26, 25, 24], that are published or submitted for publication in several journals and conference proceedings, or they are in preparation phase.

The dissertation is organized as follows:

Section 1 presents an overview of some basic concepts related to classification in general. The different types of classification of data, performance measures for assessing the quality of classification models and some examples of application are presented. The choice of document representation has

a profound impact on the quality of the classifier so different types of the text document representation are described as well as many problems and challenges that arise. The different document collections in English (Reuters-21578 and 20-Newsgroups), Chinese (Tancorp-12), Arabic (Mesleh-10) and Serbian (Ebart-3) that will be used for text classification are also presented. This section ends with a philosophical view of the classification process.

Lexical resources for Serbian [17] that have been developed within the Human Language Technologies Group at the Faculty of Mathematics, University of Belgrade are described in Section 2. They contain integrated morphological, syntactic and semantic information that can be used to improve classification accuracy of text documents in Serbian, one of the morphologically rich languages. This section describes the Serbian language corpora, system of electronic morphological dictionaries of Serbian and the lexical-semantic network, the Serbian wordnet, as well as the various natural language processing tools.

Section 3 provides a comprehensive coverage of the most important machine learning techniques used for classification task, and their application in this domain. Decision Tree methods, Distance- and Rule-based methods, Statistical methods, Neural Networks and Support Vector Machine methods are described.

New classification methods are presented in Section 4. In the case of the n-gram based method, a new n-gram weighting factors scheme is introduced. Weighting factors, which are associated with n-grams in category profiles, reflect importance of n-grams for the corresponding category with respect to other categories. In this way, n-gram with higher frequency that belongs to a smaller number of categories has a greater significance for the corresponding category. This was resulting in a two new variants of basic method: first based on modification of dissimilarity measures and second based on modification of category profiles. The second method is based on sets of well-chosen concepts from the Serbian wordnet, assigned to the corresponding categories. Each set includes literals from chosen concepts, and literals from all other concepts that are in syntactic or semantic relationship with chosen concepts. Category assignment function is defined for an test document as the maximum number of occurrences of all literals associated to the set of the chosen concepts assigned to the category, maybe filtered by domains.

Section 5 reports on experimental results of presented new classification methods. On Serbian corpus, comparison of the new variants of n-gram based method with the basic method using byte-, character-, and word-level n-grams, is presented. Only for byte-level n-grams, basic method and its modifications are tested on English, Chinese, and Arabic document collections, thus demonstrating, at the same time, language-independence of the

technique. Appendix 1 additionally presents experimental results obtain by basic n-gram method and its modifications, for all datasets and all dissimilarity measures. Method based on the Serbian wordnet is tested only on Serbian corpus.

A comparison of the results obtained by the methods presented in this dissertation with results of other classification methods is given in Section 6.

Section 7 concludes the dissertation with some discussion of the potential significance of obtained results and some directions for future work.

Key words: Text Categorization, Natural Language Processing, n-grams, Wordnet

Scientific field : Computer Science

Scientific subfield : Text processing

Sadržaj

1. Klasifikacija podataka

1.1 Uvod u klasifikaciju

Klasifikacija predstavlja preslikavanje podataka u predefinisani skup klasa (kategorija) koje su unapred poznate. Ulazni podatak u proces klasifikacije je skup slogova koji se nazivaju još i primeri ili instance. Svaki slog je oblika (x, y) gde je x skup atributa a y specijalni atribut određen za oznaku klase. Ulazni podaci se obično dele na *podatke za učenje* i *podatke za testiranje*. Na osnovu podataka za učenje (skupa slogova sa poznatim x i y), zadatak je pronaći klasifikator (klasifikacioni model, funkciju pripadnosti klasi) koji preslikava svaki skup atributa x u neku od predefinisanih klasa sa oznakom y. Cilj je dodeliti podatke (skup slogova sa poznatim x i nepoznatim y) što je moguće preciznije nekoj od klasa. Na osnovu podataka za testiranje se vrši određivanje tačnosti modela korišćenjem neke od mera evaluacije [75].

Podaci se mogu klasifikovati ručno, ali to predstavlja dugotrajan i skup proces. Mogućnosti i dostupnost brzih računara doveli su do toga da je automatsko klasifikovanje postao ključni pristup efikasnom organizovanju i obradi velike količine podataka.

Osamdesetih godina prošlog veka klasifikatori za automatsku klasifikaciju podataka konstruisali su se ručno. Klasifikatore su konstruisali stručnjaci uz pomoć eksperata iz domena sadržaja podataka. Klasifikator se sastojao od skupa ručno definisanih pravila za svaku klasu. Veliki nedostatak takvog pristupa bila je njegova nefleksibilnost. Prilikom svakog ažuriranja skupa klasa bilo je potrebno ponovo intervenisati i prilagoditi klasifikator a u slučaju promene ulaznog skupa slogova na osnovu kojih se izračunava klasifikacioni model, bilo je potrebno ponoviti ceo postupak. Tipičan primer sistema za izradu takvih klasifikatora je sistem CONSTRUE [29].

Devedesetih godina prošlog veka počeo je da se razvija novi pristup, koji je vremenom postao dominantan. Taj novi pristup omogućila je disciplina *mašinskog učenja* [52]. Tehnikama mašinskog učenja klasifikator se generiše automatski, "učenjem" karakteristika klasa na osnovu ulaznog skupa slogova odnosno skupa podataka za učenje pridruženih svakoj klasi. To su podaci

koji su ručno klasifikovani u klase od strane eksperta iz domena. Nakon procesa učenja (treniranja, podučavanja), klasifikator najčešće automatski generiše skup pravila koja treba da zadovoljava podatak za testiranje da bi bio klasifikovan u određenu klasu. Proces klasifikacije je nadgledan proces učenja (eng. supervised) budući da je vođen, odnosno nadgledan znanjem o klasama koje se stiče na osnovu podataka za učenje. Prednosti pristupa mašinskog učenja u odnosu na ručnu konstrukciju klasifikatora su evidentne. Tehnikama mašinskog učenja se ne konstruiše klasifikator, već automatski generator klasifikatora. Dakle, ako je skup klasa ažuriran ili postupak klasifikacije želi da se primeni na sasvim novi skup podataka za učenje, novi klasifikator se može generisati automatski, bez intervencije eksperata iz domena. Klasifikatori generisani tehnikama mašinskog učenja postižu impresivne rezultate, pa ovaj pristup ni kvalitativno ne zaostaje za tehnikama ručne izrade klasifikatora.

1.1.1 Vrste klasifikacije

U zavisnosti od broja klasa, klasifikacija može biti:

- *Binarana*, kada su definisane samo dve moguće klase.
- *Višeklasna*, kada je definisano više mogućih klasa.

U zavisnosti od toga da li se klase mogu preklapati ili ne, klasifikacija može biti:

- *Jednoznačna* (eng. single-label), kada jednom podatku može biti dodeljena tačno jedna klasa.
- *Višeznačna* (eng. multi-label), kada jednom podatku može biti dodeljena jedna, ni jedna ili više klasa, odnosno klase se mogu preklapati.

Klasifikacija može biti i:

- *Čvrsta* (eng. hard), kada se donosi binarna odluka (0 ili 1) da li podatak pripada određenoj klasi ili ne.
- *Meka* (eng. soft), kada se podatku pridružuje vrednost između 0 i 1 i time ocenjuje mera pripadnosti klasi.

Ukoliko se u toku procesa klasifikacije klase posmatraju samostalno bez ikakve strukture koja definiše odnose između njih, tada se radi o nehijerarhijskoj klasifikaciji. Kada broj različitih klasa, ili broj podataka unutar

jedne klase, postane jako velik, javljaju se problemi tačnog i efikasnog pretraživanja i upravljanja podacima na nivou jedne klase. U tom slučaju, klase se najčešće organizuju u stablolike strukture i uvodi se hijerahijska struktura među klasama [72]. Primer hijerarhijske klasifikacije je Yahoo[1] hijerarhija.

Dakle, prema strukturi koja definiše odnose među klasama, razlikuje se:

- *Hijerarhijska* klasifikacija.
- *Nehijerarhijska* klasifikacija.

Klasifikacija može biti još i:

- *Podatkovno orijentisana*, kada se za odabrani podatak pronalaze sve klase gde bi se on mogao svrstati.
- *Klasno orijentisana*, kada se za odabranu klasu pronalaze svi podaci koji joj pripadaju.

Razlika između ove dve vrste klasifikacije je bitna jer nisu uvek skupovi svih podatka i svih klasa dostupni. Podatkovno orijentisana je pogodnija kada podaci postaju dostupni jedan po jedan (na primer klasifikacija e-mailova), dok je klasno orijentisana pogodnija pri dodavanju novih klasa u skup postojećih klasa nakon što je deo podataka već klasifikovan (na primer klasifikovanje veb stranica).

Kada se završi proces klasifikacije, važno je dobro proceniti kakvog je kvaliteta ta klasifikaicija.

1.1.2 Procena kvaliteta klasifikacije

Veoma je važno proceniti koliko je dobro klasifikator uspeo da generalizuje problem na osnovu podataka za učenje. Jedan od problema koji se mogu javiti jeste *problem previše prilagođenog modela* (eng. overfitting). Ovaj problem se javlja kada se generisani klasifikator ponaša dobro na podacima za učenje, ali podbacuje na novim podacima za testiranje. Dakle, problem nije dobro generalizovan. To se obično javlja kada se šum koji postoji u ulaznim podacima predstavi kao bitan element.

Za testiranje se mora koristiti nezavisan skup podataka (podaci koji nisu korišćeni za učenje). Proces evaluacije se sastoji u poređenju unapred poznate klase sa onom koju je predložio klasifikator. Time se dobijaju ispravno i neispravno klasifikovani podaci [75].

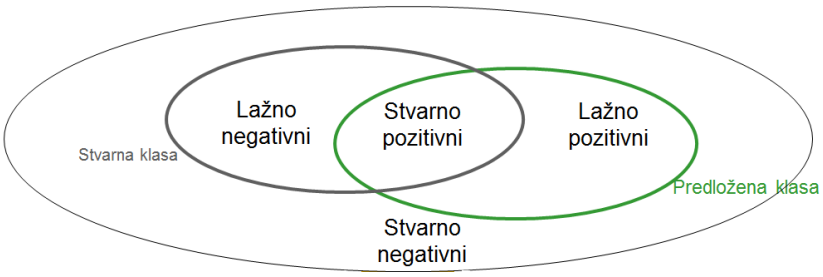
Mogući ishodi kad je u pitanju binarna klasifikacija su (videti slike 1.1 i 1.2):

[1] www.yahoo.com

- Stvarno pozitivni, SP (eng. true positives, TP)
- Stvarno negativni, SN (eng. true negatives, TN)
- Lažno pozitivni, LP (eng. false positives, FP)
- Lažno negativni, LN (eng. false negatives, FN).

		stvarna klasa	
		da	ne
predložena klasa	da	stvarno pozitivni	lažno pozitivni
	ne	lažno negativni	stvarno negativni

Slika 1.1: Mogući ishodi kod binarne klasifikacije.



Slika 1.2: Mogući ishodi kod binarne klasifikacije.

Razlikuju se dva tipa evaluacije procesa klasifikacije:

- Na nivou jedne klase.
- Na nivou više klasa.

Kada se posmatra evaluacija na nivou jedne klase, mogu se definisati sledeće mere za procenu kvaliteta klasifikacije [2]:

- *Preciznost* koja ocenjuje tačnost klasifikacije odnosno koliki procenat primera za testiranje je ispravno klasifikovan. Izračunava se po formuli:

$$Preciznost = \frac{SP}{SP + LP}$$

pri čemu $SP + LP$ predstavlja ukupan broj primera koji su dodeljeni predloženoj klasi.

- *Odziv* (pokrivanje, eng. recall) koji ocenjuje koliko je model uspešan u pokrivanju klase odnosno koliko primera za testiranje iz date klase (ili klasa) klasifikator može da prepozna:

$$Odziv = \frac{SP}{SP + LN}$$

pri čemu $SP + LN$ predstavlja ukupan broj primera koji pripadaju stvarnoj klasi.

- *F-mera* koja predstavlja kombinaciju preciznosti i pokrivanja u jednoj meri kao njihovu harmonijsku sredinu:

$$F - mera = \frac{2 * Preciznost * Odziv}{Preciznost + Odziv}$$

- *Tačnost* (eng. accuracy) koja je korisna u slučajevima kada su klase iste ili slične veličine:

$$Tacnost = \frac{SP + SN}{SP + SN + LP + LN}$$

- *Stepen greške* (eng. error rate) koja predstavlja procenat pogrešno klasifikovanih primera:

$$Stepen\ greske = \frac{LP + LN}{SP + SN + LP + LN}$$

- *Stepen razilaženja* (eng. fallout) koja predstavlja procenat pogrešno prepoznatih primera:

$$Stepen\ razilazenja = \frac{LP}{LP + SN}$$

Kvalitet klasifikacije može da bude i globalnog karaktera, kada se posmatra na nivou ne samo jedne, već više klasa. Tada se vrši usrednjavanje mera po svim klasama. To može da bude urađeno na dva načina:

- *Makro-prosek*, gde se svakoj klasi pridaje isti značaj:
  - Izračunava se vrednost mere za svaku od klasa pojedinačno.
  - Vrši se usrednjavanje tih vrednosti po broju klasa.
- *Mikro-prosek*, gde se favorizuju klase koje sadrže veći broj dokumenata:
  - Izračunavaju se vrednosti za SP, SN, LP i LN za svaku klasu pojedinačno.
  - Izračunavaju se vrednosti $\hat{SP}$, $\hat{SN}$, $\hat{LP}$ i $\hat{LN}$ kao sume svih SP, SN, LP i LN za sve klase, redom.
  - Izračunava se vrednost mere za dobijene sumirane vrednosti $\hat{SP}$, $\hat{SN}$, $\hat{LP}$ i $\hat{LN}$.

U procesu evaluacije kvaliteta klasifikacije, može se primeniti i postupak *n-unakrsnih validacija* (eng. n-cross validation) [52]. Ovaj postupak se sastoji iz sledećih koraka:

- Raspoloživi skup klasifikovanih podataka na slučajan način se deli na skupove za učenje i testiranje (na primer, u odnosu $9:1$ ili $4:1$, ređe $1:1$)
- Vrši se primena algoritma za učenje na skupu za učenje i procenjuje se kvalitet naučenog klasifikatora na skupu za testiranje.
- Ovaj postupak (koraci 1 i 2) se ponavlja n puta (obično 10 puta), i potom se izračunava srednja vrednost posmatrane ocene kvaliteta (npr. srednja vrednost preciznosti).

Performanse klasifikatora mogu da se prikažu i pomoću *matrice konfuzije* (eng. confusion matrix). Svakoj od klasa koje se koriste u procesu klasifikacije dodeljuje se po jedna vrsta i jedna kolona u matrici. Polja u koloni predstavljaju broj instanci koje je klasifikator dodelio klasi a polja u vrsti predstavljaju broj instanci koje zaista pripadaju toj klasi. Elementi na dijagonali predstavljaju broj instanci koje su ispravno klasifikovane. Primer matrice konfuzije u slučaju binarne klasifikacije (dve klase C_1 i C_2) prikazan je tabelom 1.1. Ako se klasifikacija posmatra na nivou jedne klase, na primer C_1, onda broj a predstavlja broj stvarno pozitivnih (SP), b broj lažno pozitivnih (LP), c broj lažno negativnih (LN) a d broj stvarno negativnih (SN) instanci (videti sliku 1.2). Matrica konfuzije može biti prikazana i za slučaj klasifikacije na više od dve klase.

	Prava klasa		
		C_1	C_2
Predložena klasa	C_1	a	b
	C_2	c	d

Tabela 1.1: Matrica konfuzije u slučaju binarne klasifikacije.

1.1.3 Primena klasifikacije

Klasifikacija ima mnogo različitih primera primene, a neki od njih su:

- Klasifikacija dokumenata:
  - Klasifikacija novinskih članaka (na primer, prema rubrikama sport, ekonomija, politika i drugo).
  - Klasifikacija naučnih članaka.
  - Klasifikacija veb stranica.
  - Klasifikacija e-mailova (na primer, na spam ili ne-spam).
- Poslovne primene:
  - Klasifikacija ispravnosti kreditnih kartica.
  - Procena kreditne sposobnosti.
  - "Profilisanje" kupaca ili korisnika usluga (na primer, kao lojalnog ili nelojalnog).
- Primene u bioinformatici – svaki gen/protein može da se predstavi kao vektor dužine n, gde je svaka dimenzija vrednost nekog eksperimenta ili neka promenljiva koja definiše aktivnost gena. Koristi se za predviđanje "ponašanja" novih gena.
  - Klasifikovanje po osnovu lokalizacija gena/proteina.
  - Klasifikovanje po osnovu super-familije kojoj bi gen/protein mogao da pripada.
  - Otkrivanje homologa.
  - Preslikavanje gena/proteina na skup funkcija i drugo.
- Primene u medicini:
  - Klasifikacija ćelija tumora kao benignih ili malignih.

- Primene u astronomiji:
  - Klasifikacija nebeskih objekata na zvezde i galaksije, na bazi slika dobijenih teleskopskim osmatranjem.
  - Klasifikacija galaksija prema njihovom obliku (elipsaste, sočivaste, spiralne i nepravilne).
  - Klasifikacija galaksija prema fazama formacije (rana, srednja i kasna).

1.2 Klasifikacija dokumenata

Klasifikacija dokumenata ili tekstova prema sadržaju (eng. content based) je problem koji je postao veoma aktuelan od druge polovine prošlog veka naovamo. Ovaj period karakteriše se veoma brzim rastom dostupnih tekstualnih dokumenata u digitalnom obliku i sve većom potrebom za brzim pretraživanjem podataka. Smatra se da je preko 80% dostupnih informacija sačuvano u tekstualnom obliku. Većina informacija je zapisana prirodnim jezikom, odnosno jezikom koji koriste ljudi za svakodnevnu komunikaciju. Za očekivati je da će tehnologije automatske obrade podataka zapisanih prirodnim jezikom biti među vodećim tehnologijama.

Formalno, klasifikacija dokumenata može da se definiše na sledeći način [68]:

Definicija 1.1 *Neka je* $C = \{c_1, ..., c_{|C|}\}$ *skup predefinisanih klasa i* $D = \{d_1, ..., d_{|D|}\}$ *skup dokumenata. Klasifikacija dokumenata je proces određivanja nepoznate funkcije*

$$\Phi : D \times C \to \{T, NT\}$$

koja može imati logičke vrednosti T *(tačno) i* NT *(netačno). Vrednost* T *funkcije* Φ *pridružena paru* (d_j, c_i) *označava da dati dokument* d_j *pripada klasi* c_i*, dok vrednost* NT *označava da dati dokument* d_j *ne pripada klasi* c_i*. Formalnije, cilj je da se pronađe funkcija*

$$\check{\Phi} : D \times C \to \{T, NT\}$$

koja što je moguće bolje aproksimira nepoznatu funkciju Φ*. Ova funcija* $\check{\Phi}$ *zove se još i klasifikator, klasifikacioni model ili klasifikaciona funkcija.*

Celokupna klasifikacija se vrši samo na osnovu sadržaja dokumenta odnosno podataka dobijenih iz samog dokumenta. Nazivi klasa nisu od značaja i predstavljaju samo simboličke oznake (labele) koje ne utiču na klasifikaciju.

1.2.1 Predstavljanje dokumenta

Prvi korak u procesu klasifikacije je predstavljanje dokumenta na način takav da ga računar može razumeti. Dokument bi trebalo da bude predstavljen tako da obuhvati informacije sadržane u dokumentu što je moguće bolje, kako bi se dokumenti sličnog sadržaja mogli poistovetiti a različitog razlikovati. Pokazalo se da način predstavljanja dokumenta ima veoma veliki uticaj na rad klasifikatora, posebno na sposobnost generalizacije [31]. Modeli predstavljanja dokumenta zavise od nivoa na kom se vrši analiza teksta. Što je viši nivo analize teksta, više se informacija dobija ali se time jako povećava složenost automatskog dobijanja takvih podataka. Iskustvo je pokazalo da redosled reči u rečenici, rečenična struktura ili razne semantičke odrednice ne doprinose mnogo boljim rezultatima algoritama za klasifikaciju [31].

Analiza teksta može da se izvrši na sledećim nivoima [31]:

1. Nivo fragmenta reči – dobijaju se morfološke informacije.
2. Nivo reči – dobijaju se leksičke informacije.
3. Nivo skupa reči – dobijaju se sintaksne informacije.
4. Semantički nivo – dobijaju se informacije o značenju teksta.
5. Pragmatički nivo – dobijaju se informacije o značenju teksta s obzirom na kontekst.

Nivo fragmenta reči: Umesto celih reči, kao atributi se mogu koristiti delovi ili fragmenti reči. Jedan od takvih primera su n-grami. N-gram se obično definiše kao niska karaktera dužine n. Na primer, ako se radi o reči "slovo", 4-grami su "_slo", "slov", "lovo", "ovo_". Prednosti ove metode su: zaštita od grešaka u pisanju, nezavisnost od jezika i drugo.

Nivo reči: Dokument se predstavlja kao vektor svih reči (atributa) koje se u njemu pojavljuju bez obzira na redosled reči u dokumentu. Ovakav pristup se često naziva "vreća reči" (eng. bag of words). Moguće je dokument predstaviti ne kao vektor svih, već kao vektor nekih odabranih reči (eng. "bag of terms", "bag of names", "bag of ...") a mogu biti izostavljene i neke "funkcijske" reči (npr. pomoćni glagoli, predlozi, zamenice, itd.). Iskustvo je pokazalo da reči same za sebe nose dovoljno informacija za uspešnu klasifikaciju dokumenata.

Nivo skupa reči: Da bi se dobila veća količina informacija, moraju se posmatrati i neke sintaksne zakonitosti među rečima. Na primer, ako se izraz "Metoda podržavajućih vektora" posmatra kao celina, dobija se više informacija nego ako se posmatra kao skup od tri nepovezane reči. Ovakvi izrazi se prepoznaju upoređivanjem sa gotovom listom složenih izraza ili statističkom metodom kojom se posmatra učestalost zajedničkog pojavljivanja reči u tekstu [31]. Pokazalo se da ovakav pristup samo malim delom doprinosti procesu klasifikacije pa se trud koji je potrebno uložiti u takvu implementaciju, ne isplati [33].

Semantički nivo: Postoje različite metode za obuhvatanje semantike ili značenja nekog teksta. Jedna od njih je latentno semantičko indeksiranje[14] kao i druge metode zasnovane na korišćenju predikatske logike i semantičkih mreža kao na primer wordnet [51]. Mana ovog pristupa je velika složenost i neupotrebljivost za većinu aplikacija.

Pragmatički nivo: Osnovni problem pri radu s prirodnim jezikom je taj što kontekst u kome je nešto izrečeno ima vrlo veliki uticaj na značenje izrečenog. Obuhvatiti kontekst prirodnog jezika znači opisati njegovu strukturu. Bogatstvo prirodnog jezika onemogućava stvaranje sistema koji bi ga u potpunosti obuhvatio. Pitanje je kako statistički obuhvatiti osećaj da znamo mnogo o nečemu što će neko reći pre nego što on išta i kaže [4]?

Izbor atributa

Jedan od osnovnih problema koji se javlja kod predstavljanja dokumenata jeste predimenzionisanost, odnosno veliki broj atributa kojim se dokument predstavlja. Dva su osnovna motiva za redukciju skupa atributa kojima se predstavlja neki dokument [31]:

- Zaštita od previše prilagođenog modela (eng. overfitting).
- Visokodimenzionalni prostor neprimenjiv za mnoge algoritme.

Postoje dva pristupa u procesu izbora atributa:

- *Selekcija atributa* (eng. feature selection), kod koga se na osnovu određenog znanja bira podskup početnog skupa atributa, kao "najkorisnijih" među njima. U ove metode ubrajaju se:
  - Eliminacija stop reči.
  - Frekvencija dokumenata, FD (eng. document frequency, DF).

  - Informativnost atributa, IA (eng. information gain, IG).
  - Uzajamna informacija, UI (eng. mutual information, MI).
  - χ^2 test.
  - Snaga atributa, SA (eng. term strength, TS) i drugo.

- *Ekstrakcija atributa* (eng. feature extraction), kod koga se od početnog skupa atributa stvara nov skup kao proizvod atributa starog skupa. U ove metode ubrajaju se:
  - Svođenje na koren reči.
  - Lematizacija.
  - Tezaurus.
  - Latentno semantičko indeksiranje.
  - Konceptualno indeksiranje i drugo.

Eliminacija stop reči: Jedan od načina da se izvrši eliminacija irelevantnih atributa iz skupa jeste eliminacija stop reči. Stop reči su one reči koje ne nose informaciju ni o jednoj klasi. To su najčešće veznici, predlozi i prilozi. Karakteriše ih visoka frekvencija pojavljivanja. Primer za srpski jezik su: i, na, u, već, ali i drugo. Stop reči se filtriraju na osnovu pripremljene liste reči ili na osnovu učestalosti pojavljivanja [19].

Frekvencija dokumenata: Frekvencija dokumenata je broj dokumenata iz kolekcije u kojima se pojavljuje određeni atribut. Postupak se sastoji u izračunavanju frekvencija dokumenata svih atributa i odbacivanju onih atributa čija frekvencija dokumenata ne prelazi određeni prag. Čak i za niske pragove, dimenzionalnost prostora se prilično redukuje. Ova metoda polazi od pretpostavke da retki atributi nisu dovoljno informativni ili nisu dovoljno pouzdani kako bi se uzeli u obzir pri klasifikaciji. Naime, šum u tekstu (na primer, pogrešno napisane reči) pojavljuje se u obliku niskofrekventnih reči. Dakle, odbacivanjem niskofrekventnih atributa, smanjuje se dimenzionalnost i odstranjuje se šum. Ova metoda se najlakše implementira i složenost je gotovo linearna u odnosu na broj dokumenata.

Informativnost atributa: Informativnost atributa, IA (eng. information gain, IG) predstavlja rezultat statističkog testa koji se računa na osnovu prisutnosti određenog atributa u dokumentu. On meri količinu informacije o klasi kojoj pripada dokument, a koju nam donosi saznanje o prisutnosti određenog atributa u tom dokumentu.

Neka je $C = \{c_1, ...c_m\}$ skup od m različitih klasa. Informativnost atributa t definiše se sledećim izrazom:

$$IA(t) = -\sum_{i=1}^{m} p(c_i)log(p(c_i)) + p(t)\sum_{i=1}^{m} p(c_i|t)log(p(c_i|t)) + \\ +p(\bar{t})\sum_{i=1}^{m} p(c_i|\bar{t})log(p(c_i|\bar{t})) \tag{1.1}$$

Za svaki atribut izračunava se njegova informativnost po formuli 1.1 i odbacuju se svi oni atributi čija je informativnost niža od nekog unapred određenog praga. Kako bi se dobila vrednost za IA, moraju se prvo izračunati pripadajuće uslovne verovatnoće, a zatim entropija ($E = -\sum_{i=1}^{m} p_i log p_i$). Izračunavanje uslovnih verovatnoća ima vremensku složenost $O(N)$ i prostornu složenost $O(VN)$ gde je N broj dokumenata a V broj atributa. Izračunavanje entropije ima vremensku složenost $O(Vm)$ [94].

Uzajamna informacija: Mera uzajamne informacije, UI (eng. mutual information, MI) predstavlja međusobnu zavisnost dve promenljive. Definiše se sledećim izrazom:

$$UI(t, c) = log\frac{P(t \wedge c)}{P(t) \times P(c)} \tag{1.2}$$

pri čemu oznaka t predstavlja atribut a oznaka c kategoriju. Vrednost $UI(t, c)$ jednaka je nuli kada su t i c nezavisni. Da bi se koristila ova mera pri izboru atributa, moraju se dobiti vrednosti određenog atributa uzimajući u obzir sve klase. To se uglavnom radi tako što se izračuna ili prosečan rezultat ovog atributa po klasama ili se uzme u obzir samo maksimalna vrednost koju atribut postiže na jednoj od klasa:

$$UI_{avg}(t) = \sum_{i=1}^{m} P(c_i)UI(t, c_i)$$

$$UI_{max}(t) = \max_{i=1}^{m}\{UI(t, c_i)\}$$

Vremenska složenost izračunavanja uzajamne informacije iznosi $O(Vm)$, slično kao i kod mere informativnosti atributa.

Nedostatak ove metode leži u činjenici da daje prednost retkim atributima. To se može videti na primeru izraza 1.3 koji je ekvivalentan izrazu 1.2 gde je vidljivo kako kod atributa koji imaju jednake uslovne verovatnoće viši

rezultat postiže onaj koji se ređe pojavljuje. Može se zaključiti da rezultati nisu uporedivi za atribute koji se jako razlikuju u frekvencijama pojavljivanja.

$$UI(t,c) = logP_r(t|c) - logP_r(t) \tag{1.3}$$

χ^2 test: χ^2 test je statistički test koji meri odstupanje od očekivane raspodele, ako se pretpostavi da je pojavljivanje atributa nezavisno od klase [19]. Jednostavnije rečeno, χ^2 test meri nedostatak nezavisnosti između atributa t i klase c. Ovaj test dat je izrazom 1.4 gde A označava broj zajedničkih pojavljivanja t i c, B označava broj pojavljivanja t bez c, C označava broj pojavljivanja c bez t, D predstavlja broj dokumenata u kojima se nisu pojavili ni t ni c dok je N ukupan broj dokumenata u skupu.

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{1.4}$$

Vrednost $\chi^2(t,c)$ je jednaka nuli kada su t i c nezavisni.

Kada se χ^2 test koristi kao mera pri izboru atributa, vrši se sumiranje rezultata po klasama. Računanjem χ^2 testa između svakog pojedinog atributa i određene klase dolazi se do izraza za računanje χ^2 testa po klasama:

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i)\chi^2(t,c_i)$$

$$\chi^2_{max}(t) = \max_{i=1}^{m}\{\chi^2(t,c_i)\}$$

Računanje χ^2 testa ima kvadratnu prostornu složenost, slično kao i mere za informativnost atributa i uzajamnu informaciju. Kao statistički test poznat je po tome da je nepouzdan za male vrednosti koje su česte pri klasifikaciji dokumenata. Ove male vrednosti najčešće su posledica atributa koji se retko pojavljuju.

Snaga atributa: Ova metoda predstavlja verovatnoću da se atribut pojavi u usko povezanim dokumentima. Dokumenti u skupu za učenje izdvajaju se u parove ako je njihova sličnost iznad određene vrednosti. Snaga atributa se računa pomoću procene uslovne verovatnoće da se atribut pojavljuje u drugom delu para povezanih dokumenata ako se pojavljuje u prvom delu para.

Neka x i y predstavljaju proizvoljan par različitih ali povezanih dokumenata, i neka je t atribut. Definicija snage atributa t data je izrazom:

$$s(t) = P(t \in y | t \in x) \tag{1.5}$$

Ova mera je potpuno drugačija od prethodno navedenih mera. Zasniva se na grupisanju dokumenata uz pretpostavku da su dokumenti koji sadrže veliki broj sličnih reči, i sami slični. Takve reči sadrže veliku količinu informacije. Ova metoda ne uzima u obzir povezanost između atributa i klasa. Po tome je slična metodi učestalosti dokumenata a znatno se razlikuje od, na primer, informativnosti atributa ili mere uzajamne informacije.

Za razliku od metoda za selekciju atributa, kojima se vrši izbor podskupa početnog skupa atributa, metode za ekstrakciju atributa od početnog skupa atributa stvaraju nov skup atributa.

Svođenje na koren reči: Ovom metodom smanjuje se broj atributa tako što se poistovećuju atributi sa istim korenom. Koren reči je ono što ostane od reči kada joj odstranimo prefiks i sufiks (ukoliko ih reč ima). U engleskom jeziku, na primer, "connect" je koren s kojim se poistovećuju reči: "connected", "connecting", "connection" i "connections". Smisao svođenja na koren reči je u tome da se više leksički različitih reči predstavi atributom koji je zapravo koncept koji te reči predstavljaju. Tu međutim može doći do greške jer je netačno tvrditi da reči sa istim korenom uvek ukazuju na isti koncept ako se zna da i reč sama po sebi dobija svoje puno značenje tek u kontekstu.

Jedan od najčešće korišćenih algoritama za određivanje korena reči je onaj kojeg je dao Porter [60] a koji koristi kontekstno osetljiva pravila za odbacivanje sufiksa. Porterov algoritam je jednostavan i dobro funkcioniše za engleski jezik nije morfološki bogat jezik [35]. Korisnost ove metode dolazi do izražaja kod morfološki bogatijih jezika kao što je srpski jezik.

Lematizacija: Lematizacija je proces sličan procesu određivanja korena reči. Razlika je u tome što se ovde, umesto na koren, reč svodi na njen osnovni oblik ili lemu. Ovaj proces je dosta teži od procesa određivanja korena reči jer se reč mora tačno identifikovati.

Tezaurus: Dimenzionalnost prostora može da se smanji i tako što se na osnovu tezaurusa pronalaze svi sinonimi (različiti literali sa istim značenjem) koji se zatim izjednačavaju, odnosno zamenjuju predstavnikom grupe. U svojoj osnovnoj formi, ovakav se tezaurus sastoji od liste reči bitnih za dati domen, a svaka od tih reči nosi sa sobom skup reči istog ili sličnog značenja

[2]. Ova metoda najbolje rezultate daje ako se radi o specifičnom domenu ili ako se koristi sa kontrolisanim rečnikom.

Tezaurus može da sadrži različite relacije među rečima. Osim sinonima koji se vode pod relacijom ekvivalencije, tu se ubrajaju i relacije poput "opštije od" i "specifičnije od" [31]. Primer takvog tezaurusa je wordnet [51] koji je razvijen za različite jezike pa i srpski [26].

Latentno semantičko indeksiranje: Dobijanje informacija obično se vrši upoređivanjem literala (reči) iz dokumenata sa literalima iz upita. Međutim, leksičke metode mogu biti neprecizne kada se radi s korisničkim upitima, jer postoji mnogo načina na koje se može prikazati željeni koncept. Zbog toga, literali iz upita ne moraju odgovarati literalima iz dokumenata relevantnih za taj upit. Takođe, postoje literali koji imaju višestruka značenja (polisemija), tako da literali iz upita mogu odgovarati literalima iz dokumenata koji nisu relevantni za taj upit. Latentno semantičko indeksiranje nastoji da zaobiđe ove probleme izbegavajući direktan rad sa literalima (rečima) iz dokumenata time što koristi statistički dobijene indikacije o konceptima koje ti literali predstavljaju [3].

Latentno semantičko indeksiranje vrši preslikavanje vektora atributa u podprostor smanjene dimenzije, koristeći metodu dekompozicije singularnih vrednosti. Izračunava se ortogonalna transformacija koordinatnog sistema a nove koordinate odgovaraju novim atributima. Bira se samo s najvećih singularnih vrednosti, pa rezultujući podprostor ima manju dimenziju uz najmanje moguće odstupanje. Broj singularnih vrednosti s koje treba izabrati kako bi se dobili najbolji rezultati je u početku nepoznat, a do njega se može doći unakrsnom validacijom [31].

Ova metoda dobro se nosi sa problemom sinonima. Međutim, problem polisemije se ne može rešiti ovom metodom, jer je svaki pojam predstavljen kao jedna tačka u prostoru. Pojmovi sa više značenja će biti predstavljeni prosečnom vrednošću tih značenja koja mogu biti vrlo različita [14].

Konceptualno indeksiranje: Metode za dobijanje informacija zasnovane na redukciji dimenzionalnosti, kao latentno semantičko indeksiranje, pokazale su poboljšanje kvaliteta dobijenih informacija jer uspevaju da obuhvate i prikrivena značenja reči. Međutim, metoda latentnog semantičkog indeksiranja nije u stanju da iskoristi informacije o klasi pojedinih dokumenata u postupku redukcije dimenzionalnosti pa se u zavisnosti od strukture skupa podataka efekti mogu jako razlikovati.

Konceptualno indeksiranje zasniva se na određivanju grupa sličnih dokumenata na osnovu kojih se zatim određuju ose nižedimenzionalnog prostora.

U slučaju nadziranog procesa, dokumenti se prvo dele u skupove prema pripadnosi odgovarajućoj klasi. Zbog toga će na početku biti onoliko novih osa koliko je i različitih klasa. Postupak se dalje sprovodi tako što se ove grupe dokumenata dele na podgrupe, a zatim se nove ose usmeravaju prema njima. Bitno je primetiti da i nakon podela grupa u podgrupe, svaka novostvorena grupa sadrži samo dokumente jedne klase [36].

Pridruživanje težina atributima

Pridruživanje težina atributima može da se posmatra kao blagi oblik selekcije atributa, jer dok selekcija atributa u potpunosti uklanja neke atribute, pridruživanje težina samo menja njihov uticaj [31].

Pridruživanje težina u većini slučajeva se zasniva na tri standardne pretpostavke:

1. Retki atributi nisu manje važni od čestih.
2. Višestruka pojavljivanja nisu manje važna od jednostrukih pojavljivanja.
3. Dugački dokumenti nisu važniji od kratkih.

S obzirom na ove pretpostavke, težine atributa se obično sastoje od tri komponente:

- Komponente dokumenta.
- Komponente kolekcije.
- Normalizacijske komponente.

Uobičajene mere za komponentu dokumenta zadate su tabelom 1.2 [31].

Intuitivno, veća važnost se pridaje atributima koji se češće pojavljuju. Zbog toga se kao osnovna mera i navodi frekvencija atributa. Frekvencija atributa $FA(w_i, d_j)$ definiše se kao broj pojavljivanja reči (atributa) w_i u dokumentu d_j.

Sami podaci o frekvenciji atributa ne mogu osigurati dobre rezultate pogotovu ako je njegova frekventnost pojavljivanja podjednako visoka u svim dokumentima kolekcije. Kako bi se takvim atributima mogla pridružiti manja težina, uvodi se komponta kolekcije koja se bazira na vrednosti frekvencije dokumenata. Frekvencija dokumenata $FD(w_i)$ je mera koja se definiše kao broj dokumenata u kojima se pojavljuje atribut (reč, term) w_i. Prihvatljivi indikator vrednosti atributa kao diskriminatora dokumenata može se zadati kao inverzna funkcija frekvencije dokumenta tog atributa, IFD (eng, inverse

1.0	za izraze koji su prisutni u dokumentu, mera ima vrednost 1, a inače 0.
$FA(w_i, d)$	frekvencija atributa, broj pojavljivanja atributa w_i u dokumentu d.
$0.5 + 0.5\frac{FA(w_i,d)}{max_j FA(w_j,d)}$	normallizovana frekvencija atributa (smanjena je razlika između čestih i retkih izraza.)

Tabela 1.2: Uobičajene mere za komponentu dokumenta.

document frequency, IDF). Jasno je da su najpoželjniji atributi koji imaju veliku frekvenciju atributa a malu frekvenciju dokumenata. Tako se ove prethodne mere mogu kombinovati u meru koja tvrdi da su najbolji atributi oni koji su frekventni u individualnim dokumentima ali se retko pojavljuju u ostalom delu kolekcije. To je FA-IFD (na engleskom,TF-IDF) mera koja se izračunava kao proizvod mera FA i IFD. Uobičajene mere komponente kolekcije navedene su u tabeli 1.3 [31].

1.0	ignoriše se frekvencija dokumenta.
$IFD = log\frac{\|D\|}{FD(w_i)}$	$\|D\|$ je ukupan broj dokumenata u kolekciji (atributi koji se pojavljuju u više dokumenata će dobiti niže težine).
$log\frac{\|D\|-FD(w_i)}{FD(w_i)}$	probabilistički inverz frekvencije dokumenata.

Tabela 1.3: Uobičajene mere za komponentu kolekcije.

Kod skupova sa većim varijacijama u dužinama dokumenata bitna je i treća, normalizacijska komponenta. Uobičajene mere za normalizacijsku komponentu prikazane su u tabeli 1.4.

Ako se koristi veliki broj atributa za predstavljanje dokumenata, jasno je da će vektori dužih dokumenata biti popunjeniji i time imati veću šansu da upoređivanjem sa atributima iz upita budu označeni kao relevantni. Međutim svi relevantni dokumenti trebalo bi da budu tretirani isto bez obzira na dužinu (videti tabelu 1.4), pa je stoga zadatak normalizacijske komponente

1.0	bez normalizacije.
$\frac{1}{\sqrt{\sum_j x_j^2}}$	normalizacija rezultujućeg vektora na dužinu 1 u 2-normi
$\frac{1}{\sum_j x_j}$	normalizacija rezultujućeg vektora na dužinu 1 u 1-normi.

Tabela 1.4: Uobičajene mere za normalizacijsku komponentu.

da izjednači dužine vektora dokumenata [66].

U [13] izložen je i korak dalje pa se vrši nadzirano pridruživanje težina. Umesto da se gleda raspodela atributa po celoj kolekciji gledaju se razlike u raspodelama kod pozitivnih i negativnih primera iz kolekcije. Iako se govori o povećanju efikasnosti, ne može se govoriti o konstantnoj superiornosti nad FA-IFD metodom.

Na osnovu svega ovde izloženog može da se zaključi da izbor načina na koji će dokument biti predstavljen u procesu klasifikacije predstavlja veliki izazov. Osim izbora odgovarajućih atributa i karakteristika za predstavljanje dokumenata kao i dodele težina tim atributima, problem može da predstavlja i to što su neki atributi teški za generisanje (npr. eksperimentalni podaci).

1.2.2 Problemi i izazovi

Osim problema predstavljanja dokumenta, neki od izazova u procesu klasifikacije dokumenata su:

- *Brzina* kojom se gradi model i kojom se primenjuje model na nove dokumente za testiranje.
- *Velike i male klase.* Klasifikacija koja dodeljuje svima većinsku klasu u principu pravi najmanju grešku. Izazov je napraviti klasifikator za prepoznavanje "manjinskih" klasa.
- *Zamućena (eng. fuzzy) klasifikacija* ili klasifikacija sa određenim stepenom značajnosti [75].

Pre samog procesa klasifikacije, izazov je napraviti dobar skup podataka za učenje, što obično zahteva "ručnu" izradu. To je vrlo često jako skupo, sporo i podložno greškama. Zbog toga je posebno važno povesti računa o šumovima ili greškama u podacima za učenje.

Skupovi dokumenata za učenje i testiranje mogu biti napisani na različitim jezicima. U fazi selekcije i ekstrakcije atributa, često se koriste znanja

iz oblasti lingvistike i prirodnih jezika. Proces predstavljanja dokumenta u takvim situacijama, zavisan je od jezika na kome je tekst u dokumentu napisan. Obično se zahteva korišćenje rečnika i drugih jezičkih resursa i složenih alata iz oblasti obrade prirodnih jezika. Algoritmi razvijeni u svrhu predstavljanja dokumenta napisanog na, na primer, engleskom jeziku ne mogu se primeniti na dokumente napisane na nekom drugom jeziku. Svaki jezik ima svoje specifične karakteristike koje se moraju poštovati. Pre samog procesa obrade tekstualnog dokumenta, važne osobine jezika na kome je taj dokument napisan se moraju uzeti u razmatranje.

Na primer, za razliku od engleskog jezika, srpski jezik se odlikuje veoma bogatom morfologijom, bogatom rečeničnom strukturom i velikom slobodom u redosledu reči u rečenici. Takođe, koriste se dva pisma (ćirilica i latinica), pravopis je fonološki baziran i postoji pravilo o smeštanju enklitika (oblika pomoćnih glagola i ličnih zamenica, koje nemaju akcenat) u rečenici. Kod kineskog i drugih azijskih jezika, glavni problem je proces izdvajanja reči u rečenici. Kod ovih jezika reči nisu eksplicitno razdvojene belinama, pa je taj postupak prilično komplikovan. Time se prikaz dokumenta zasnovan na nivou reči (na primer "vreća reči") značajno komplikuje. Tekst na arapskom jeziku se na primer piše s desna na levo. To je jezik sa veoma složenom morfologijom. Neki samoglasnici su predstavljeni dijakritičkim znacima koji se obično ukanjaju iz dokumenta, što dovodi do velike dvosmislenosti. Vlastite imenice se ne pišu velikim slovima.

Sve to dodatno komplikuje proces klasifikacije dokumenata ali ujedno predstavlja i veliki izazov, posebno u razvijanju metoda koje su bez obzira na sve, jezički nezavisne.

1.2.3 Korpusi klasifikovanih dokumenata

Prvi korak u procesu klasifikacije jeste prikupljanje dokumenata u korpus i njihova podela na skupove za učenje i testiranje. U ovom radu su korišćeni korpusi na srpskom (Ebart-3), engleskom (Reuters-21578 i 20-Newsgroups), kineskom (Tancorp-12) i arapskom jeziku (Mesleh-10).

Ebart-3

Ebart korpus[2] predstavlja najveću digitalnu medijsku dokumentaciju u Srbiji, sa preko dva miliona novinskih tekstova iz štampanih medija koji imaju nacionalnu pokrivenost, kao i odabranih lokalnih medija, arhiviranih od početka 2003. godine naovamo.

[2] http://www.arhiv.rs/

Korisnicima su dostupne raznovrsne predefinisane pretrage tekstova, pa se svi paketi mogu pretraživati po:

- Temama – oko 900 tema podeljenih na 3 nivoa kao na primer, Industrija, Prehrambena industrija, Konditorska industrija i drugo.
- Ličnostima – koje se mogu pretraživati uz različita ukrštanja sa temama i drugim ličnostima.
- Institucijama – ekonomske, društvene, kulturne i drugo.
- Političkim strankama – sve parlamentarne i važnije neparlamentarne.
- Geografskim odrednicama – Beograd, Beč, Bečej, Beočin, Balkan, Banat... i tako do 550 različitih gradova, sela, regiona, država, planina i drugo.
- Manifestacijama – sajmovi, festivali, nagrade i drugo.
- Dokumentima – važni zakoni, odnosno sve što su novine pisale o hipotekama, koncesijama, porezima, radiodifuziji, stečaju, telekomunikacijama i drugo.
- Događajima – svi izbori, afere, veliki međunarodni skupovi, vanredno stanje i drugo.

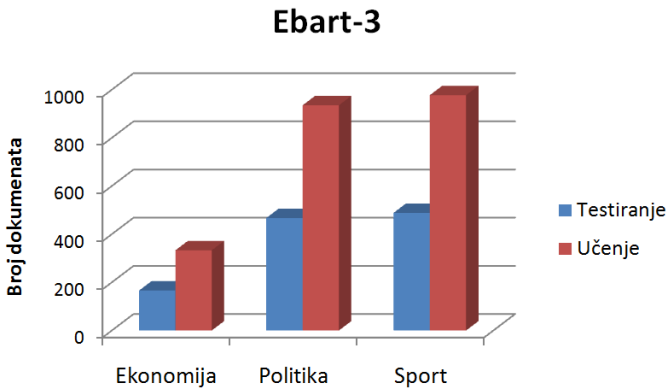
Aktuelna arhiva je klasifikovana na tematske celine po ugledu na uobičajene novinske rubrike: unutrašnja politika, spoljna politika, društvo, ekonomija, hronika i kriminal, kultura i zabava, sport, mediji, feljtoni, pisma čitalaca.

U ovom radu korišćen je podskup Ebart korpusa nazvan Ebart-3 koji predstavlja slučajno odabrane članke iz dnevnih novina "Politika" koji pripadaju rubrikama/klasama: sport, ekonomija i politika, izdatim od 2003. do 2006. godine. Svaki članak/dokument može pripadati samo jednoj klasi. Ima ukupno 3366 takvih članaka. Ovaj korpus podeljen je na podatke za učenje i podatke za testiranje u odnosu $2:1$. Slika. 1.3 prikazuje raspodelu ovog korpusa po pomenutim rubrikama/klasama i prikazuje koliko novinskih članaka pripada kojoj klasi, uzimajuću u obzir podelu na skupove za učenje i testiranje. Korpus je dostupan u tekstualnom i xml formatu.

Reuters-21578

Reuters novinska kolekcija je korpus novinskih članaka na engleskom jeziku. Najčešće se koristi Reuters-21578[3] verzija korpusa koja sadrži 21578 dokumenata, zbog čega je i dobila naziv. Korpus Reuters-21578 je naslednik korpusa

[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/



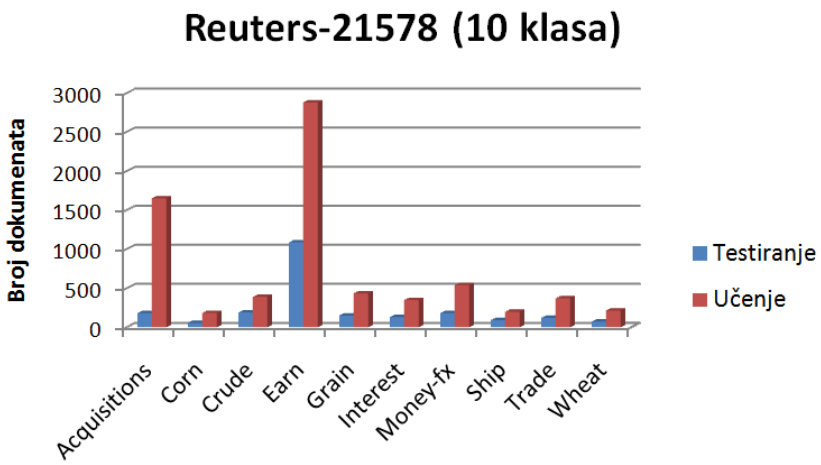
Slika 1.3: Raspodela Ebart-3 korpusa po klasama.

Reuters-22173 koji predstavlja skup novinskih članaka koje je objavila novinska agencija Reuters 1987. godine. Dokumente su iste godine prikupili i ručno klasifikovali zaposleni u Reuters Ltd. i Carnegie Group Inc. Javno dostupan akademskoj zajednici postao je 1990. godine, a dalje formatiranje kolekcije obavili su David Lewis i Stephen Harding. David Lews [48] je nakon daljeg sređivanja i izbacivanja 595 dupliranih dokumenata, 1996. godine objavio konačnu verziju kolekcije Reuters-21578 [48].

U cilju primene ovog korpusa na problem klasifikacije tekstova, potrebno je standardizovati podelu korpusa na podatke za učenje i testiranje. Postoji više predefinisanih podela na dokumente za učenje i dokumente za testiranje, ali jedna od najčešće korišćenih je ModApte (Modified Apte Split) podela u okviru koje 75% dokumenata pripada skupu za učenje a 25% skupu za testiranje. Svakom dokumentu može da se pridruži više klasa. Jedna klasa može sadržati između 1 i 2877 dokumenata u skupu za učenje, odnosno između 1 i 1066 dokumenata u skupu za testiranje. Broj klasa je 135, međutim samo se 90 klasa pojavljuje bar jednom i u skupu za učenje i u skupu za testiranje [31]. Od tih 90 mogućih klasa, najčešće se koristi samo 10 najfrekventnijih. U ovom radu je korišćena restrikcija ovog korpusa na 10 najfrekventnijih klasa.

Ovaj korpus karakteriše se neravnomernom raspodelom broja dokumenata po klasama. Tako na primer, među 10 najfrekventnijih klasa, klasa "zarada" (eng. earn) sadrži 40% svih dokumenata a 8 od ostalih 10 klasa ukupno sadrže manje od 7.5%. Na slici 1.4 je prikazana raspodela ovog korpusa restrikovanog na 10 najfrekventnijih klasa.

Kolekcija je distribuirana u 22 datoteke od kojih poslednja sadrži 578 dokumenata a ostale sadrže po 1000 dokumenata. Datoteke su pisane u SGML (Standard Generalized Markup Langage) [93] formatu.



Slika 1.4: Raspodela Reuters-21578 korpusa po klasama.

20-NewsGroups

20-NewsGroups[4] je kolekcija od oko 20000 tekstova na engleskom, ravnomerno raspoređenih na 20 različitih diskusionih grupa ili klasa, pri čemu svaka grupa odgovara različitoj temi. Ovu kolekciju je prvi prikupio Ken Lang [47].

Tri verzije ovog korpusa su javno dostupne. Najpopularnija je "bydate" verzija. U ovoj verziji su dokumenti sortirani po datumu u skup za učenje (60%) i skup za testiranje (40%), ne sadrže duplikate i ne sadrže zaglavlja kao na primer Xref, Newsgroups, Path, Followup-To, Date.

Za razliku od Reuters-21578, u okviru ovog korpusa svaki dokument pripada samo jednoj grupi/klasi. Neke grupe su veoma blisko povezane (na primer comp.sys.ibm.pc.hardware i comp.sys.mac.hardware) a neke su veoma različite (na primer misc.forsale i soc.religion.christian). U tabeli 1.5 je prikazana lista svih 20 diskusionih grupa/klasa, manje ili više podeljena prema sadržaju.

Tancorp

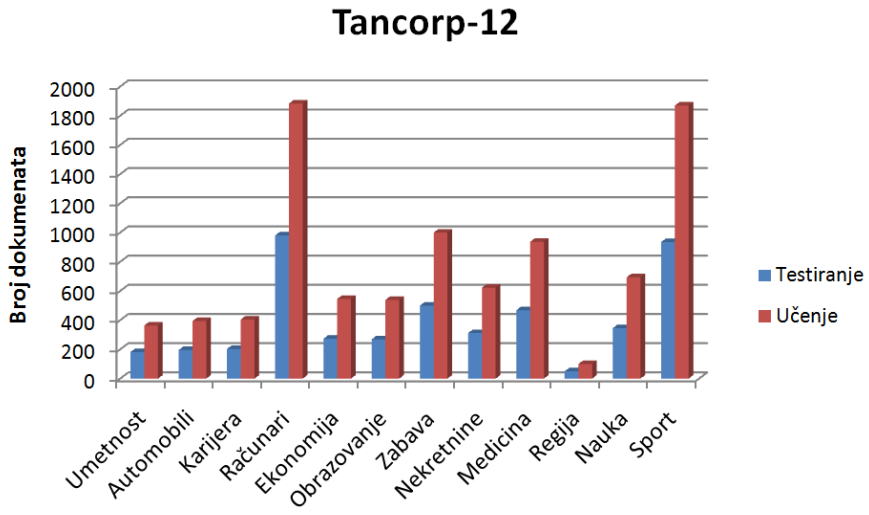
Tancorp[5] je kolekcija dokumenata na kineskom jeziku, sakupljena od strane Songbo Tan-a [76]. Korpus sadrži 14150 dokumenata hijerarhijski organizovanih. Prvi nivo hijerarhije sadrži 12 velikih klasa a drugi sadrži 60 malih klasa. Ovaj korpus može da se koristi u formi tri skupa podataka: jedan hijerarhijski korpus (TanCorpHier) i dva nehijerarhijska korpusa (TanCorp-12

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/
[5] http://www.searchforum.org.cn/tansongbo/corpus.htm

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Tabela 1.5: 20 diskusionih grupa/klasa u 20-Newsgroups korpusu

i TanCorp-60) koja redom sadrže 12 i 60 klasa. U ovom radu korišćen je Tancorp-12 korpus. Dokumenti za učenje i testiranje su slučajnim izborom podeljeni u skupove za učenje i testiranje u odnosu 2 : 1. Tancorp se kao i Reuters-21578 korpus karakteriše neravnomernom raspodelom broja dokumenata po klasama što je prikazano na Slici 1.5.

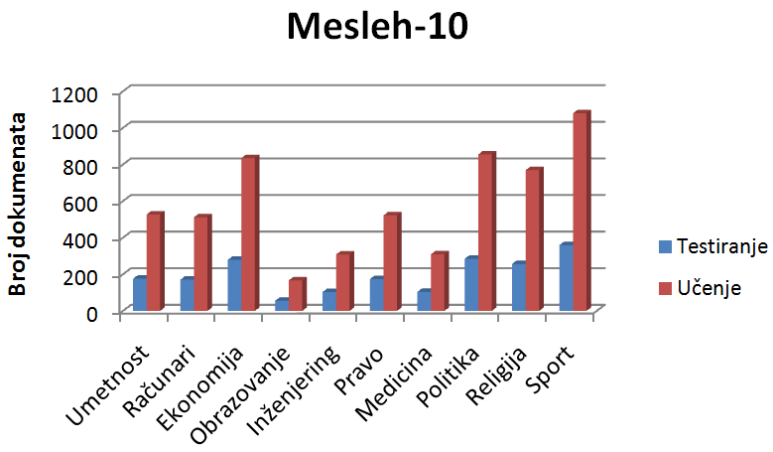


Slika 1.5: Raspodela Tancorp korpusa po klasama.

Mesleh-10

Mesleh-10 je korpus novinskih članaka na arapskom jeziku koji je predstavio Mesleh u radu [50]. Dobijen je iz javno dostupnih arhiva kao na primer Al-Jazeera, Al-Nahar, Al-hayat, Al-Ahram, Al-Dostor i drugo. Korpus sadrži

7842 dokumenata neravnomerno raspoređenih u 10 klasa. Podeljen je u skupove za učenje i testiranje u odnosu 3 : 1. Na Slici 1.6 je prikazana raspodela korpusa po klasama.



Slika 1.6: Raspodela arapskog korpusa Mesleh-10 po klasama.

1.3 Teorijske osnove klasifikacije

Da li je postojanje samostalne teorije klasifikacije u oblasti pretraživanja informacija potrebno i korisno ili nije?

Rasprava oko ovog pitanja vodi se u radovima Hjorlanda [30] i Spark Jones [34]. Složili su se da su teorije klasifikacije neadekvatno i nedovoljno razmatrane i da je dalji razvoj oblasti pretraživanja informacija otežan zbog nedovoljno razvijene samostalne teorije klasifikacije koja bi omogućila adekvatnu klasifikaciju podataka.

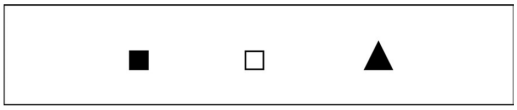
Klasifikacija predstavlja grupisanje podataka sa sličnim karakteristikama i ponašanjem prema nekim definisanim pravilima ili kriterijumima. Izbor i opis karakteristika nekog podatka prema kojima se vrši njegova klasifikacija, veoma zavise od osobe koja vrši klasifikaciju, od njenog obrazovanja, predznanja o datoj oblasti, socijalno-kulturološke osnove i teorijskog stanovišta. Postoji mnogo načina na koje se neki podaci mogu klasifikovati. Kao što je već rečeno, klasifikacija na primer može biti binarna i višeklasna, jednoznačna i višeznačna, čvrsta i meka, hijerarhijska i nehijerarhijska, podatkovno i klasno orijentisana. Na osnovu samih podataka koji se klasifikuju, nije uvek jasno koja vrsta klasifikacije treba da se primeni. Izbor vrste klasifikacije takođe zavisi od osobe koja tu klasifikaciju obavlja. Ljudi sa različitim teori-

jskim stanovištem, teže da opišu iste podatke različito. Informatika se ne bavi u osnovi istraživanjem o tome kako ljudi zapravo klasifikuju stvari. To je pre svega fokus nekih drugih polja istraživanja kao na primer psihologije i socialne antropologije. Informatika je prvenstveno fokusirana na normativnu teoriju klasifikacije i bavi se pitanjem: koji kriterijum treba izabrati za klasifikaciju dokumenata u cilju optimizacije dobijanja informacija?

1.3.1 Pragmatizam nasuprot pozitivizmu

Jones u svom radu [32] predlaže dva teorijska pristupa klasifikaciji: pozitivistički i pragmatički. Pozitivizam je pristup u kome se treba osloniti samo na činjenice, na čisto posmatranje, logičke dedukcije i formalne modele, zanemarujući pitanja koja se odnose na tumačenje značenja, ciljeva, svrhe i vrednosti.

Kako ne postoji neki opšte prihvaćen, prirodni i najbolji način za klasifikaciju datog skupa podataka, potrebno je definisati formalne kriterijume po kojima će se meriti koliko je neka klasifikacija dobra. Ako se klasifikacija izvodi sa nekom svrhom, potrebno je tačno i precizno definisati tu svrhu. Ovo se najbolje može ilustrovati primerom na slici 1.7. Data su dva kvadrata i trougao. Jedan kvadrat i trougao obojeni su crnom bojom a drugi kvadrat belom bojom. Postavlja se pitanje kako izvršiti klasifikaciju ovih podataka, po boji ili po obliku? Ni jedna od ove dve klasifikacije nije bolja od druge. Koja će od ove dve biti izabrana, zavisi isključivo od svrhe u koju se klasifikacija obavlja.



Slika 1.7: Kriterijumi za klasifikaciju.

Različiti ljudski interesi naglašavaju različite osobine podataka: na primer, farmaceuti i hemičari ističu različite osobine istog hemijskog elementa (hemičari ističu strukturne osobine a farmaceuti medicinske efekte). Štaviše, unutar svakog polja delovanja, različite teorije i "paradigme" takođe ističu različite osobine.

Ni pretraživač informacija na vebu nije neutralni alat. Dizajniranje nekog sistema treba da bude vođeno specifičnim interesima tog sistema. Na primer, sistem dizajniran za skandinavsku javnost ne treba da teži identifikovanju američkih komercijalnih veb strana već treba da bude usmeren ka identifikovanju strana koje reflektuju kulturološke i demografske vrednosti i svrhe.

Klasifikacija se dakle obavlja na osnovu znanja o njenoj svrsi, ciljevima i vrednostima, što je osnova pragmatičnog pristupa. Kod ovakvog pristupa klasifikaciji, odluke su uvek teorijski vođene, odnosno zavise od specifične teorije, pristupa i pogleda na svet koji postoji u datom domenu. Smatra se da u slučaju nepragmatičnog pristupa problemu klasifikacije, cena može biti visoka kao na primer korišćenje neadekvatne metode za klasifikaciju. Zbog svega toga, jako je važno pristupiti problemu na adekvatan način.

1.3.2 Klasifikacija i konceptualizam

Pojam klasifikacije prepliće se sa pojmovima koncepta i konceptualizma pa se tako za pojam koncepta vezuju različiti stavovi i teorije u skladu sa razlikama između pozitivizma i pragmatizma.

Različite teorije na različite načine opisuju koncept. Andersen i njegove kolege u radu [1] predstavljaju dve teorije o konceptima. Prva je takozvana *klasična teorija*, po kojoj je koncept skup pojedinačno neophodnih a zajedno dovoljnih osobina. Druga je *Kuhn-ova teorija* nazvana još i *prototip teorija*, po kojoj podaci u sličnim klasama ne treba da imaju više od "porodičnih" sličnosti sa svojim bližnjima, samim tim i da koncepti koji pripadaju sličnim klasama su porodično slični koncepti. Kuhn odbacuje specifičan skup osobina kao osnovu za članstvo u klasi, čime odbacuje tradicionalni stav da su koncepti i klase definisani u smislu potrebnih i dovoljnih svojstava. Posledica ove Kuhn-ove teorije je da identifikovanje članova kao pripadajućih ili ne nekoj klasi, ne mora biti zasnovano na istom skupu osobina. Na primer, koncept "hrana" unutar neke kulture predstavlja koncept koji unutar te kulture indukuje šta je jestivo i razlikuje ga od onog što nije jestivo. Pojam "jestivo" može biti definisan različito od kulture do kulture.

Problem sinonima i homonima je takođe povezan sa problemom klasifikacije. U procesu klasifikacije, opisi samih podataka mogu sadržati izvestan stepen sinonima i homonima, sto znači da se isti podaci i njihove osobine mogu označiti različito a različiti podaci mogu da se označe slično.

Zaključak je da kvalitet klasifikacije, koja predstavlja grupisanje podataka u unapred određene klase, prema nekom kriterijumu a na osnovu nekih osobina, uglavnom zavisi od kvaliteta tog izabranog kriterijuma. Način na koji čovek posmatra neki podatak zavisi od njegovog obrazovanja, predznanja o datoj oblasti, socijalno-kulturološke osnove i teorijskog stanovišta. Pre samog procesa klasifikacije, moraju da budu razvijena značenja i moraju da budu ugrađena u koncepte, terminologije i ceo naučni komunikacioni sistem. Klasifikacija u informatici ne treba da bude bazirana na trivijalnim i naivnim opisima dokumenata, već na znanju u okviru nekog konteksta. Dokumenti koji se klasifikuju mogu da budu privrženi nekom određenom gledištu (na

primer feminističkom) ali važno je da korisnici klasifikacije imaju pristup različitim stanovištima i informacijama i to ne samo dominantnih ideologija već i nekih ređe prisutnih. Nije moguće biti potpuno neutralan ali nije prihvatljivo ni da se različita gledišta skrivaju i da se tako suzbija sposobnost korisnika da razvije sopstveno mišljenje. Zbog toga i postoji tako duboka i nezaobilazna nedoumica vezana za klasifikaciju a to je teorijska i ideološka osnova na kojoj klasifikacija treba da bude izgrađena ("neutralni" sistemi teže da budu zasnovani na dominantnom stanovištu i obično nisu viđeni kao neutralni sa spoljne tačke gledišta) [30]. Ljudi koji razvijaju alate za klasifikaciju i sprovode klasifikaciju treba da budu adekvatno obrazovani i da budu u poziciji da razlikuju teorijska stanovišta kao i da u osnovi razumeju njihov značaj i posledice. Oni treba da budu u poziciji da razumeju različite pristupe i da klasifikaciju baziraju na kompromisu između različitih teorijskih stanovišta.

2. Leksički resursi i tehnologije njihove obrade

Kompleks naučnih, tehnoloških i poslovnih informacija, u svom najznačajnijem delu obuhvata informacije predstavljene prirodnim jezikom. Organizovanje informacija, posebno njihovo organizovanje u informatičkom smislu, je svrhovito tek kada omogućava pronalaženje pohranjene informacije [85]. Prema [95], tekst čine "podaci u obliku karaktera, simbola, reči, pasusa, rečenica, tabela, ili drugačijih rasporeda karaktera, namenjenih prenošenju značenja i čija se interpretacija suštinski zasniva na poznavanju nekog prirodnog ili veštačkog jezika". Ova definicija teksta određuje tekst kao sekvenciju karaktera kojom se prenosi nameravano značenje, ali na takav način da ova informatička reprezentacija teksta ne enkodira eksplicitno čak ni informaciju o jeziku čije je poznavanje potrebno da bi se tekst protumačio. Štaviše, ovoj definiciji podjednako odgovaraju sekvencije karaktera koje čine tekst jedne strane u html-u, dokument proizveden procesorom reči ili skenirana slika tog dokumenta. Ovakva interpretativna priroda informatičkog zapisa teksta podrazumeva da će korisnik pohranjenog teksta znati da protumači enkodiranu poruku, ali nam ne govori ništa o slučaju kada se u ulozi korisnika nađe računar. U tekstu koji je definisan na ovakav način, računar ne može prepoznavati lingvističke objekte kao što su reč, rečenica ili pasus bez ugrađenog znanja o ovakvim objektima [85]. Iz tog razloga, informacija često ostaje skrivena iako je njena reprezentacija načelno saglasna sa navedenom definicijom teksta. Zbog toga je veoma značajna jezička podrška sistemima za pronalaženje informacija. U tu svrhu mogu se koristiti različiti *jezički resursi*.

Pod jezičkim resursima se podrazumevaju jezički zavisni podaci i alati koji se mogu primeniti u rešavanju problema obrade prirodnih jezika [85]. Jezički resursi sadrže širok spektar različitih lingvističkih informacija u zavisnosti od njihove prirode i funkcije kojoj su namenjeni. Posebna vrsta jezičkih resursa su *leksički resursi* koji mogu biti različitih tipova. S jedne strane, to su kolekcije tekstova kao što su morfološki ili sintaksički obeleženi korpusi, a sa druge,

elektronski leksikoni i formalne gramatike. Razvoj ove druge grupe resursa je spor i skup jer se oni konstruišu ručno u određenom teorijskom okviru. Kao nadoknadu za sporost u izgradnji, leksikoni i gramatike omogućavaju značajno složenije obrade od obrada zasnovanih na korpusu [85]. U zavisnosti od broja jezika na koje se odnose, leksički resursi mogu biti podeljeni na jednojezične, dvojezične i višejezične. Kod dvojezičnih i višejezičnih leksičkih resursa, reči među različitim jezicima mogu biti povezane ili ne. Moguće je izgraditi leksički resurs koji se sastoji od različitih rečnika istog jezika. Na primer, jedan rečnik se može odnositi na uopštene reči a jedan ili više rečnika se mogu odnositi na reči koje pripadaju specijalnim domenima.

Leksički resursi za srpski jezik se razvijaju u okviru Grupe za jezičke tehnologije na Matematičkom fakultetu Univeziteta u Beogradu (Grupa) već duži niz godina, tako da je danas na raspolaganju veliki broj različitih resursa, razvijenih u značajnom obimu [88]. Pored korpusa srpskog jezika, kao i višejezičnih paralalenih korpusa, od posebnog značaja su sistem morfoloških rečnika srpskog jezika (SMR) razvijenih u okviru mreže RELEX [11], kao i semantička mreža za srpski jezik (*srpski wordnet* — SWN) razvijena u okviru međunarodnog projekta Balkanet.

Pored već pomenutih resursa, u Grupi se koriste i razvijaju i *sistemi konačnih automata predstavljenih grafovima*, koji se u lingvističkim softverima koriste za formalizaciju lingvističkih fenomena i za obradu (parsiranje) teksta, a pored njih, i dvojezične, *paralelne liste*, kao pomoćni resurs pri pretraživanju i prevođenju. Konačno, Grupa učestvuje i u razvoju višejezične *ontologije vlastitih imena* [85], u okviru Prolex [58] projekta, organizovane oko koncepta vlastitog imena, kao jedinstvenog koncepta u različitim jezicima. U višejezičnom kontekstu, opis vlastitih imena ne može da se svede samo na elektronski rečnik, zbog kompleksnosti semantičkih veza koje ih povezuju. U ovoj mreži je srpski predstavljen sa oko 2.000 vlastitih imena (pre svega, imena država i njihovih glavnih gradova) uključujući opis specifične semantike vlastitih imena. Za opis pojedinih fenomena izgrađene su *lokalne gramatike* [83] koje omogućavaju da se opiše struktura i dodeli gramatičko značenje nizovima prostih reči koji se ne mogu eksplicitno navesti u rečniku. Neki primeri lokalnih gramatika su gramatike za prepoznavanje nizova brojeva zapisanih rečima ili gramatike za prepoznavanje datuma u tekstu [40]. Ovaj resurs se može koristiti i za obeležavanje nestrukturiranog teksta, na primer XML-etiketama, u kaskadama konačnih transduktora [87].

Pored različitih formata resursa, poseban problem su i različiti kodni rasporedi koji su se vremenom javljali u resursima, počev od takozvanog aurora zapisa, u kome su slova ć, č, š, ž, đ, dž, lj i nj kodirana ASCII karakterima cx, cy, sx, zx, dx, dy, lx i nx, preko ISO 8859-2 i ISO 2 8859-5 koda, pa do Unicode-a.

2.1 Korpus srpskog jezika

Srpski jezik raspolaže korpusom savremenog srpskog jezika koji je nastao 2002, na inicijativu Ljubomira Popovića i Duška Vitasa [59] i koji omogućava različita jezička istraživanja preko veba polazeći od izabrane kolekcije tekstova. Ovaj korpus se može konsultovati i kroz sistem Unitex[1] koji obradu korpusa efikasno povezuje sa sistemom elektronskih rečnika i lokalnih gramatika, ali se ne može koristiti preko veba. Unitex omogućava formulisanje kompleksnih upita bilo u obliku regularnih izraza ili u obliku rekurzivnih mreža prelaza [87].

2.1.1 Višejezični paralaleni korpusi

Pod paralelizovanim tekstom ili bitekstom se obično podrazumevaju tekst i njegov prevod (ili njegovi prevodi) predstavljeni na takav način da je između elemenata njihove logičke strukture uspostavljena eksplicitna veza, na primer, na nivou pasusa ili rečenica. Srpski jezik raspolaže sa dva značajna paralelizovana korpusa. Jedan je *Intera-korpus*[2], koji sadrži po milion reči srpskog i engleskog, uparen na nivou rečenice i u TMX-formatu[3]. Srpski tekstovi u ovom korpusu su lematizirani na nivou prostih reči sa obeleženom vrstom reči. Drugi je *francusko-srpski literarni korpus*, koji se sastoji prvenstveno od izbora dela francuske književnosti prevedenih na srpski [86]. Korpus sadrži oko 1.300.000 reči na francuskom i 1.100.000 na srpskom i uparen je na nivou rečenice. Za većinu tekstova u korpusu je ručnom kontrolom obezbeđeno jednoznačno uparivanje originala i prevoda. Paralelni korpusi, pored kontrastivnih istraživanja, našli su primenu i u eksperimentima u oblasti automatskog prevođenja i automatskoj ekstrakciji termina [84].

2.2 Elektronski rečnik

Elektronski rečnik ili e-rečnik je rečnik koji je predstavljen u elektronskoj formi i koji je namenjen ekskluzivno automatskoj transformaciji teksta (za razliku od mašinski čitljivih rečnika koji su namenjeni korišćenju od strane čoveka). Između ostalog, njegova glavna svrha je korišćenje u procesu obrade prirodnog jezika. Ovo znači da elektronski rečnik sadrži i one informacije koje omogućavaju da se razreše problemi segmentacije, morfološke, a delimično i sintaksičke i semantičke obrade teksta [85]. Model elektronskog rečnika koji

[1] http://igm.univ-mlv.fr/ũnitex/
[2] http://www.clarin.eu/intera-corpus
[3] http://www.lisa.org/Translation-Memory-e.34.0.html

se pokazao podesnim za srpski, ali i za druge slovenske jezike, razvijen je polazeći od metodologije koja je nastala u okviru mreže RELEX [11].

2.2.1 Sistem morfoloških rečnika srpskog jezika

Sistem morfoloških rečnika *SrpMD*[4] srpskog jezika [40, 88], sastoji se od nekoliko osnovnih delova: *DELAS*, koji predstavlja rečnik prostih reči u osnovnom obliku (prostih lema), *DELAC*, rečnika složenih reči (kontingentnih niski prostih reči) i *DELAF*, rečnika oblika prostih reči, kao i morfoloških gramatika koje omogućavaju prepoznavanje "nepoznate" reči, odnosno reči koja nije prepoznata na osnovu postojećih rečnika. Aktuelni obim SrpMD obuhvata oko 80.000 prostih reči iz kojih je generisan rečnik DELAF sa preko 3.000.000 oblika prostih reči. Svakom zapisu u rečniku prostih lema (DELAS) pridružena je informacija o vrsti reči i, ako je potrebno, kod flektivne klase, precizan opis promene reči. Elementima rečnika DELAS se mogu dodati morfosintaktičke, sintaksičke ili semantičke kao i informacije o izgovoru. Tako je, na primer, pridev "devojcyin" u rečniku DELAS zapisan sa [70]:

```
devojcyin, A1+Pos+Ek (1)
```

što znači da se radi o pridevu koji pripada flektivnoj klasi A1, koji je prisvojan (+Pos), ekavskog izgovora (+Ek). Informacije iz sistema rečnika SrpMD mogu se pomoću sistema Intex [11] koristiti za formulisanje kompleksnih upita za pretraživanje tekstova. Na primer, upitom: `<A+Pos-Ek>` će se dobiti svi prisvojni pridevi u tekstu koji ne pripadaju ekavskom izgovoru. Zapisi u DELAS rečniku mogu sadržati i derivacione relacije kojima se povezuju reči koje pripadaju istom derivacionom gnezdu. Ovaj tip informacija se razdvaja znakom podvake (_). Na primer:

```
devojcyin, A1+Pos+Ek_N=4ka (2)
devojka, N618+Hum+Ek_A=2cyin
```

Informacije koje se nalaze iza podvlake u pridevu "devojcyin" (A) povezuju ga sa imenicom "devojka" (N) tako što se poslednja četiri karaktera "cyin" zamene sa "ka". U drugom redu pokazano je kako se, na sličan način, imenica "devojka" povezuje sa pridevom "devojcyin". Sem toga, morfosintaktičkim informacijama (ispred kojih stoji znak plus), može se opisati i tip derivacione veze između dve leme. U primeru (2), iz oznake +Pos vidi se da je pridev "devojcyin" prisvojni pridev imenice "devojka". Ove informacije iz rečnika DELAS mogu se koristiti za lematizaciju tekstova na osnovu proizvoljno izabrane leme iz jednog derivacionog gnezda, uz pomoć konačnih transduktora.

[4] http://korpus.matf.bg.ac.rs/SrpMD/

2.2.2 Rečnik vlastitih imena

Sastavni deo sistema elektronskih rečnika predstavljaju rečnici vlastitih imena. Rečnik vlastitih imena nastao je kao deo Prolex projekta [58] u cilju projektovanja i implementacije višejezičnog rečnika vlastitih imena i relacija među njima. Od 1996. godine u okviru ovog projekta razmatrana su vlastita imena (posebno toponimi, lična imena, inostrana lična imena i enciklopedijski pojmovi) i istaknuta je potreba da se sva vlastita imena povežu zajedno. Kreirana je višejezična baza vlastitih imena, nazvana *Prolexbase*, sa lingvističkim informacijama korisnim u procesu obrade prirodnih jezika.

Za reprezentaciju vlastitih imena i relacija među njima korišćen je XML zbog prednosti koje ima kao što su omogućena integracija i razmena informacija među lingvističkim podacima [6].

Kao što se može primetiti, rečnik se sastoji iz dva dela, jednog koji se odnosi na relacije i drugog koji se odnosi na jezike koji su u njemu definisani.

Prvi deo, koji se odnosi na relacije, ima koren u elementu *relationships* i sastoji se iz:

- Liste elemenata tipa *pivot* koji predstavljaju apstraktnu notaciju za definisanje opštih relacija između vlastitih imena.
- Liste elemenata tipa *predication* koji povezuju dva pivota sa određenim iskazom određenog jezika.
- Liste elemenata tipa *type* pri čemu je svaki tip koren hijerarhijski uređenog skupa tipova i
- Elementa tipa *Wordnet* koji beleži veze sa *wordnet*-om.

Svaka od ovih listi elemenata može biti prazna.

Element *pivot* ima jedinstveno određen identifikator i on predstavlja koncept koji mora postojati u bar jednom od definisanih jezika. Zbog toga u okviru svakog *pivot* elementa mora postojati element *concept* za koji je definisan jezik u okviru atributa *language* i vlastito ime u okviru atributa *prolexeme*. Svaki pivot, odnosno koncept, može da se odnosi na samo jedno vlastito ime u okviru jednog jezika ili na više njih ali svako u okviru različitog jezika. Sama vlastita imena su opisana u drugom delu XML dokumenta.

Element *prediction* definiše vezu između dva pivota (koncepta) koja je definisana u okviru nekog jezika. Svaki element ovog tipa ima obavezno element *pReference* koji definiše jezik i iskaz ili vezu među vlastitim imenima određenog jezika. Ovaj element u okviru *prediction* odgovara elementu *concept* u okviru elementa *pivot*. Može se primetiti da atribut *language* u okviru elemenata *concept* i *pReference* ima veliku ulogu kod aplikacija koje služe za

prevođenje iz jednog jezika u drugi. Zaista, na ovaj način pristup vlastitom imenu iz jednog u drugi jezik obavlja se automatski.

Rečnik vlastitih imena ima široku primenu u procesu obrade prirodnih jezika kao što su automatsko prevođenje, izdvajanje informacija, višejezično poravnanje teksta i tako dalje.

2.3 Wordnet

Wordnet koji je danas poznat kao *prinstonski wordnet* (PWN) razvili su Džordž Miler i njegov tim sa ciljem da se koristi kao jedna vrsta mentalnog leksikona u okviru psiholingvističkih projekata [18]. U okviru tradicionalnih rečnika, leksički pojmovi su alfabetski uređeni i za svaki od njih je data definicija za svako od mogućih značenja. Za razliku od toga, kod *wordnet*-a su sve reči kojima se može izraziti neki pojam grupisane zajedno u skup sinonima ili sinset (eng. synset, synonymous set) predstavljajući tako jedan koncept. PWN predstavlja skup približno 100.000 koncepata povezanih semantičkim relacijama u semantičku mrežu. Projekat *EuroWordnet* (EWN) [88] [89] je dao projektu *Wordnet* novu dimenziju uvodeći višejezičnost u semantičku mrežu. Vokabulari sedam evropskih jezika su prvo organizovani na sličan način kao PWN, a zatim međusobno povezani preko takozvanog međujezičkog indeksa (eng. Inter-Lingual-Index — ILI). *BalkaNet* je projekat koji je od septembra 2001. do avgusta 2004. finansirala Evropska komisija [79]. Cilj ovog projekta je razvoj poravnatih semantičkih mreža tipa *wordnet* za balkanske jezike, i to bugarski, grčki, rumunski, srpski i turski, kao i proširenje mreže za češki koja je početno bila razvijana u okviru projekta EWN. Osnovni cilj *BalkanNet* projekta je razvoj savremenih jezičkih resursa za balkanske jezike koji bi omogućili nov način pristupa informacijama koje potiču iz balkanskih jezika. Osim toga, cilj ovog projekta bio je i proširenje semantičke mreže koja je uspostavljena u okviru projekta EWN balkanskim jezicima. Svrha ovakvog proširenja je da se ojača saradnja balkanskih zemalja sa članicama Evropske unije. Kao glavne aktivnosti u okviru *BalkaNet* projekta treba istaći, pre svega, razvoj mreža *wordnet* za balkanske jezike pojedinačno (bugarski, grčki, rumunski, srpski, turski i češki) i njihovo povezivanje sa postojećom leksičkom bazom EWN. Ove glavne aktivnosti su planirane i sprovedene sinhronizovano, što znači da su jednojezičke mreže izgrađene nad zajednički dogovorenim osnovnim skupovima od 8516 koncepata već prisutnim u PWN-u. To su takozvani "bazični koncepti". Izvan ovih osnovnih skupova "bazičnih koncepata", za svaki pojedinačni jezik mreža se razvijala nezavisno, ali u okvirima koje je postavio PWN. Ovakav prisput razvoju mreže je postavio specifične probleme. Naime, tokom rada na razvoju mreže

često su se postavljala sledeća pitanja: da li su koncepti jezički zavisni ili ne, da li su obrasci za leksikalizaciju koncepata univerzalni, da li je struktura prinstonske mreže valjana i za druge jezike i da li je skup semantičkih relacija koje su u njega ugrađene dovoljan za sve jezike [90]. Premda je rad na razvoju zasebnih mreža za balkanske jezike često davao potvrde za negativan odgovor na ova pitanja, nije se odustalo od prethodno utvrđenog postupka. U odsustvu srpskog rečnika i dvojezičnog englesko/srpskog rečnika u elektronskoj formi, prevod koncepata iz PWN u *srpski wordnet* (SWN) je rađen ručno. Iz tog razloga, postavilo se pitanje validnosti SWN-a. Korišćenje jednojezičnih i višejezičnih korpusa u cilju provere validnosti sinsetova u SWN-u dovelo je do dodavanja novih i uklanjanja postojećih literala iz nekih sinsetova [54][42].

Kako se mreže tipa *wordnet* danas razvijaju pre svega za informatičke potrebe, tako se i osnovna primena ovih mreža za balkanske jezike vidi u njihovoj ugradnji u informatičke primene koje su zasnovane na prirodnojezičkoj obradi. Mogu se koristiti za klasifikaciju dokumenata ili višejezičko pretraživanje, uključivanje u pretraživačke mašine (obeležja domena) - poboljšanje usluga mašina za pretraživanje za većinu balkanskih jezika, konceptualno indeksiranje veb stranica, npr. Alexandria (Memodata, Lingvistička podrška i usluge za veb korisnike)[5]. Postojanje višejezične baze sa međusobno poravnatim konceptima u ovim slučajevima je od suštinskog značaja.

Ipak, da bi se prevazišli uočeni problemi, svi partneri na projektu su se dogovorili da se kao jedan od rezultata rada na ovom projektu ugradi i skup koncepata koji su specifični za balkanske jezike [39].

2.3.1 Srpski wordnet

Srpski wordnet[6] je leksičko-semantička mreža srpskog jezika [43, 55]. Struktura SWN je u osnovi ista kao struktura PWN i organizovana je preko čvorova i relacija između tih čvorova. Kao što je već rečeno, ovi čvorovi se u wordnetu nazivaju sinsetovi i predstavljaju zapravo skupove reči koje u nekom kontekstu imaju isto značenje. Svaka reč u sinsetu predstavljena je niskom karaktera ili literalom, za kojom sledi značenje tog konkretnog literala u konkretnom sinsetu. Ovo rešenje se zasniva na pristupu koji se koristi u klasičnim rečnicima govornog jezika, gde jednoj reči odgovara više mogućih značenja, koja se na poseban način obeležavaju. Kako u *wordnet*-u neka reč može imati više značenja, to može biti član više različitih sinsetova.

Ova baza je podeljena u delove prema vrstama reči, i to prema imenicama,

[5] http://www.memodata.com/2004/en/alexandria/
[6] http://korpus.matf.bg.ac.rs/SrpWN

glagolima, pridevima i prilozima. Prema stanju SWN-a iz januara 2013. godine[7], u tabeli 2.1 je predstavljen broj sinsetova za odgovarajuću vrstu reči.

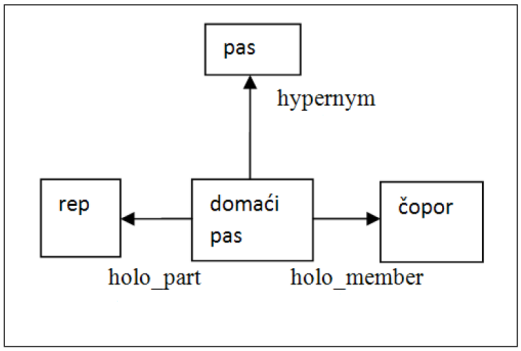
Vrsta reči	Srpski wordnet
Imenice	14765
Glagoli	2104
Pridevi	1380
Prilozi	117
Ukupno	18366

Tabela 2.1: Raspodela sinsetova (koncepata) u SWN-u prema vrsti reči, u januaru 2013. godine

Imenički deo baze je organizovan kao hijerarhijska mreža imeničkih čvorova koja se uspostavlja na osnovu postojanja relacije *podređenosti* (eng. hyponym) i *nadređenosti* (eng. hypernym) između pojmova koje ti čvorovi predstavljaju. Za jedan pojam kažemo da je podređen drugom pojmu ako poseduje sva svojstva koja poseduje i nadređeni pojam, ali ima i neka specifična svojstva. Relacija podređenosti i nadređenosti svakako nije jedina koja se uspostavlja između pojmova. Značajne su i relacije *deo-celina* (eng. holo_part) i *član-celina* (eng. holo_member). Na primer, sinset {pas:C1x, pseto:1, domacxi pas:1} je povezan relacijom nadređenosti sa sinsetom {*pas* : *C*1}, sinset {rep:2a} je povezan relacijom *deo-celina* sa sinsetom {pas:C1x, pseto:1, domacxi pas:1} a ovaj sinset je povezan relacijom *član-celina* sa sinsetom {cyopor:1}. Na slikama 2.1 i 2.2 su prikazani idealizovani model i XML reprezentacija wordnet-a koji ilustruju ovaj primer. Relacija *antonimije* (eng. near_antonym) se uspostavlja između imeničkih sinsetova i njome se povezuju (približno) suprotni pojmovi.

Druga značajna grupa relacija uspostavljenih između sinsetova u wordnetu jesu one koje povezuju pojmove koji se leksikalizuju različitim vrstama reči. Važna relacija koja povezuje imeničke i pridevske sinsetove jeste *biti u stanju-stanje nečega* (eng. be_in_state), a jedan primer predstavlja sinset {cyistocxa:1} (stanje čiste osobe; bez prljavštine ili drugih nečistoća) koji je povezan sa pridevskim sinsetom {cyist:1a} (koji je bez prljavštine ili nečistoće ili ima naviku da bude čist). Relacija antonimije je česta među pridevima, pa je sinset {cyist:1a} povezan relacijom *skoro suprotan* sa sinsetom {prlxav:1} (koji na sebi ima prljavštinu ili nečistoću). Ovaj sinset je, pak, u vezi sa imeničkim sisnetom {prlxavsxtina:1, necyistocxa:1} (stanje nekoga

[7]http://korpus.matf.bg.ac.rs/SrpWN/SrpWN_01_2013.pdf



Slika 2.1: Deo *wordnet*-a (idealizovani model)

ili nečega što nije čisto) preko relacije *biti u stanju-stanje nečega* , dok je ovaj sinset opet povezan relacijom *skoro suprotan* sa sinsetom {cyistocxa:1}. Ako ovome pridodamo i relacije koje se uspostavljaju između glagolskih sinsetova, kao što je relacija *uzrokuje-uzrokovan* (eng. verb_group) koja, na primer, povezuje sinsetove {uspraviti:1, podignuti:3} (postaviti u uspravan položaj) i {stajati:1a} (biti u uspravnom položaju) jasno je da je u wordnetu uspostavljena gusta mreža između čvorova [41]. U tabeli 2.2 je prikazana lista svih relacija i broj njihovog pojavljivanja u SWN, prema stanju iz januara 2013.

2.4 Tehnologije obrade leksičkih resursa

Predstavljeni različiti tipovi resursa nastajali su tokom dužeg vremenskog perioda pa su samim tim razvijani u okviru različitih projekata i stoga neminovno unutar različitih konceptualnih i tehnoloških okvira. Iako je Grupa ulagala velike napore da stepen koherentnosti i standardizovanosti resursa bude što veći, određena mera heterogenosti nije mogla da se izbegne. Ovi leksički resursi razvijeni su na osnovu sasvim različitih modela pa samim tim sadrže i različite vrste leksičkih informacija. Sve to je motivisalo članove Grupe da pristupe razvoju softverskih sistema, odnosno softverskih alata, koji će sa jedne strane olakšati dalji razvoj i održavanje leksičkih resursa, a sa druge strane će olakšati njihovu integraciju. Time je omogućeno znatno lakše obavljanje niza zadataka vezanih za obradu tekstova u elektronskom obliku.

```
<SYNSET>
      <ID>ENG30-02083346-n</ID>
      <SYNONYM>
            <LITERAL>pas</LITERAL>
      </SYNONYM>
      <DEF>Bilo koji od raznovrsnih
           sisara koji obicyno imaju
           dugu nxusxku i kandye.
      </DEF>
      <POS>n</POS>
</SYNSET>

<SYNSET>
      <ID>ENG30-02084071-n</ID>
      <SYNONYM>
            <LITERAL>pas</LITERAL>
            <LITERAL>pseto</LITERAL>
            <LITERAL>domacxi pas</LITERAL>
      </SYNONYM>
      <DEF>Pripadnik Canis familiaris,
           srodan vuku, pripitomlxen od
           preistorijskog doba;
           postoje mnoge rase.
      </DEF>
      <POS>n</POS>
      <ILR>ENG30-02083346-n
            <TYPE>hypernym</TYPE>
      </ILR>
      <ILR>ENG30-07994941-n
            <TYPE>holo_member</TYPE>
      </ILR>
</SYNSET>
<SYNSET>
      <ID>ENG30-02158846-n</ID>
      <SYNONYM>
            <LITERAL>rep</LITERAL>
      </SYNONYM>
      <DEF>Upadlxivo oznacyen ili oblikovan
           zadnxi deo.</DEF>
      <POS>n</POS>
      <ILR>ENG30-02084071-n
            <TYPE>holo_part</TYPE>
      </ILR>
</SYNSET>

<SYNSET>
      <ID>ENG30-07994941-n</ID>
      <SYNONYM>
            <LITERAL>cyopor</LITERAL>
      </SYNONYM>
      <DEF>Grupa zxivotinxa koje love.</DEF>
      <POS>n</POS>
</SYNSET>
```

Slika 2.2: Deo *wordnet*-a (XML reprezentacija)

Relacije u SWN	Broj pojava
hypernym	16866
eng_derivative	2926
subevent	78
category_domain	738
near_antonym	762
verb_group	170
also_see	226
causes	64
holo_part	1560
holo_member	3879
holo_portion	209
usage_domain	15
be_in_state	287
similar_to	244
derived	662
particle	10
derived-gender	38
derived-pos	45
derived-vn	2
region_domain	82

Tabela 2.2: Raspodela relacija u SWN-u u januaru 2013. godine

2.4.1 LeXimir

Poseban značaj među ovim softverskim alatima ima radna stanica koja je dobila naziv *LeXimir*[8] i koja omogućava sihronizovano korišćenje raznorodnih resursa [45, 71]. Prethodna verzija ovog alata imala je akronim *WS4LR* (eng. Workstation for Lexical Resources) [44]. Kao nadogradnja ovog alata u Grupi za jezičke tehnologije, razvijena je veb aplikacija *VebRanka*[9] čiji je cilj bio da omogući razvoj i korišćenje jezičkih resursa za srpski jezik i na vebu.

LeXimir sadrži nekoliko komponenti koje izvršavaju različite funkcije:

- *Konverzija* je komponenta koja omogućava različite vrste transformacija resursa (jedne datoteke ili skupa datoteka) koje mogu sadržati tekst, lokalne gramatike, elektronske rečnike formata DELAS i DELAF i slično. Konverzije između različitih formata resursa se uglavnom

[8] http://korpus.matf.bg.ac.rs/soft/LeXimir.html
[9] http://hlt.rgf.bg.ac.rs/vebranka/

odnose na konvertovanje iz Intex ili Unitex formata u NooJ format rečnika, grafova i regularnih izraza.

- Podsistem za održavanje *sistema morfoloških rečnika* je komponenta koja omogućava upravljanje skupom odabranih rečnika u DELA formatu koji sadrže proste ili složene reči. Odabrani rečnici mogu da budu distribuirani u više datoteka. Glavna snaga alata je mogućnost efikasnog pretraživanja i izdvajanja podskupa lema na osnovu uslova poređenja lema, vrste reči, koda flektivne klase, i sintaksičkih i semantičkih oznaka. Podsistem obezbeđuje vezu sa regularnim izrazima i sa FSA (eng. finite state automata) grafovima koji opisuju flektivna svojstva izabrane leme [40], tako da se oni mogu pregledati i korigovati ako je potrebno. Editor lema omućava formiranje leme iz početka, ili kopiranje neke postojeće slične leme koja se zatim modifikuje. Veza sa flektivnim regularnim izrazima i FSA grafovima omogućava da se odmah generišu svi flektivni oblici nove leme i tako proveri ispravnost odabranog koda flektivne klase.

- Razvoj i unapređenje wordneta je komponenta koja podržava rad sa pojedinačnim wordnet-om ali i sinhronizovan rad dva wordnet-a koji se ostvaruje preko jedinstvenog identifikatora ILI. Osim toga, sinsetovi se mogu selektovati korišćenjem različitih metoda, koje idu od jednostavnog sravnjivanja niski do kompleksnih XPath izraza za koje su pripremljeni obrasci koji odgovaraju često postavljanim zahtevima. Novi sinsetovi se mogu dodavati wordnet-u korišćenjem predefinisanih formi. Nestruktuirane, dvojezične liste pružaju pomoć i preporuku za moguće kandidate za literale novog sinseta, posebno u slučaju kada se jedan wordnet razvija sinhronizovano sa nekim već razvijenim. U ovaj modul su takođe ugrađene različite opcije za proveru konzistentnosti podataka.

- Podsistem za *interakcije sistema elektronskih rečnika i ontologija* omogućava razmenu informacija između wordnet-a i morfoloških rečnika. Naime, morfo-sintaksičke informacije iz morfoloških rečnika se mogu pridružiti literalima u sinsetu, a semantičke informacije iz sinsetova se mogu pridružiti lemama u rečnicima. Ovim modulom se mogu kreirati Intex/Unitex grafovi koji pronalaze u tekstu sve forme svih literala iz izabranog sinseta, kome se mogu dodati odabrani literali iz sinsetova koji su nadređeni izabranom.

- Značajna komponenta je i okruženje za izgradnju i eksploataciju *paralelizovanih tekstova* i konverziju TEI-formata ka drugim standard-

ima, a posebno ka TMX-u. Paralelizovani tekstovi u TMX formatu se mogu vizuelizovati na različite načine korišćenjem unapred pripremljenih XSLT skriptova. Integracija resursa u WS4LR se najbolje ilustruje kroz pretragu paralelizovanih tekstova. Korisnik kao upitni obrazac može da zada jednu nisku koja može biti lema, što znači da se pretraživanje obavlja svim flektivnim oblicima ili koncept, što znači da se pretraživanje obavlja svim literalima iz izabranih sinsetova i njihovih nadređenih pojmova. Zadovoljavanje ovih upita zahteva uključivanje praktično svih raspoloživih resursa. U izdvojenim paralelizovanim segmentima, pojavljivanja koja odgovaraju kriterijumima pretrage su osvetljena drugom bojom. Na osnovu veze između sinsetova u sinhronizovanim wordnet-ima koja se ostvaruje preko jedinstvenog identifikatora ILI, moguća je i paralelna višejezicna pretraga i označavanje nađenih reči u odgovarajućim tekstovima.

- Veb aplikacija za jezičke resurse VebRanka koristi veb servis *wsQueryExpand*[10] koji pruža različite mogućnosti proširenja upita, i omogućava ekspanziju upita na vebu koristeći Google AJAX Search API. Najveći skup predviđenih korisničkih funkcija vezan je za ekspanziju upita, bolje reći za raznovrsne mogućnosti podešavanja upita (jer sem proširivanja, omogućava i njegovo sužavanje). VebRanka kao i LeXimir, daje korisniku mogućnost da upit proširi morfološki, semantički ali i na još jedan jezik (a koji zavisi od raspoloživih resursa).

Mada se LeXimir uglavnom koristi za srpski jezik, njegovo korišćenje nije zavisno od jezika. Jedina pretpostavka je da za neki jezik resursi postoje ili da se razvijaju prema opisanim formatima i metodologijama. Sistem može paralelno da radi sa dva jezika, kombinujuci bilo koja dva jezika izabrana iz predefinisanih parametara raspoloživih resursa.

2.4.2 ACIDE

ACIDE (eng. Aligned Corpora Integrated Development Environment)[11] je integrisano razvojno okruženje za paralelizovane korpuse [80]. Motivacija Grupe za razvojem ovakvog okruženja nastala je zbog nedostatka udobnog okruženja sa grafičkim korisničkim interfejsom, koje bi u sebi objedinilo sve pojedinačne softverske komponente koje se koriste u raznim fazama pripreme paralelnih tekstova za paralelizovane korpuse. Ovo okruženje, između ostalog, obezbeđuje grafički korisnički interfejs za:

[10] http://hlt.rgf.bg.ac.rs/wsQueryExpand/service.asmx
[11] http://korpus.matf.bg.ac.rs/soft/acide.html

- Paralelizaciju.
- Vizuelizaciju paralelizovanog teksta, kontrolu i korekciju.
- Generisanje datoteka u TMX formatu.
- Razlaganje datoteka u TMX formatu na datoteke pojedinacnih jezika.
- Vertikalizaciju teksta.

ACIDE obavlja paralelizaciju korišćenjem programskih paketa XAlign i Concordancier koje je razvila laboratorija LORIA u Francuskoj[12].

2.4.3 XML baze podataka

Kako je većina leksičkih resursa predstavljena u XML formatu, obrada ovih resursa može biti sprovedena i direktno, korišćenjem XML baza podataka. XML baze podataka mogu biti XML-proširene i izvorne XML baze podataka.

Osnovne osobine XML proširenih baza podataka su:

- Koriste postojeći sistem za upravljanje bazama podataka.
- Preslikavaju XML podatke u sopstveni model pri čemu se čuvaju hijerarhija i podaci a gubi se identitet dokumenta, redosled čvorova na istom nivou i drugo.
- Namenjene su podatkovno orijentisanim (eng. data-centric) dokumentima.

Jedan od osnovnih problema koji se javlja kod XML-proširenih baza podataka je taj što polustrukturirani podaci smešteni u relacionim bazama imaju kao rezultat veliki broj nedostajućih vrednosti ili veliki broj tabela.

Izvorne XML baze podataka karakterišu se sledećim osobinama:

- XML dokument je osnovna logička jedinica, kao što je to vrsta u tabeli kod relacionih baza.
- XML dokumente smeštaju u "izvornom" obliku održavajući pri tom prirodnu drvoliku strukturu dokumenata.
- Nema zahteva za postojanjem bilo kakvog specificnog fizičkog modela skladištenja.

[12] http://led.loria.fr/

- Namenjene su dokumentno orijentisanim (eng. document-centric) dokumentima.

Veliki broj jezika je kreiran za postavljanje upita nad XML dokumentima uključujuci XML-QL, XPath, XQL, XQuery. XPath je W3C preporuka a sa pojavom XQuery postaje još popularniji. Koriste se za dobijanje i manipulisanje podacima iz XML baza podataka. Postoji veliki broj ugrađenih funkcija a korisnik ima mogućnost definisanja sopstvenih funkcija.

Xpath koristi iskaze putanja za kretanje kroz logičku, hijerarhijsku strukturu XML dokumenta. Dizajniran je da radi sa jednim XML dokumentom. Vrednost vraćena XML upitom je skup čvorova.

XQuery je jezik koji je projektovan da bude mali, da se lako implementira i da bude lako razumljiv jezik. On je nastao sa idejom da obezbedi upitni jezik koji ima istu širinu funkcionalnosti kao SQL nad relacionim bazama podataka. To je funkcionalni jezik u kome je svaki upit iskaz. Iskazi u XQuery-u upadaju u 6 širokih tipova:

- Iskazi putanje.
- Konstruktori elemenata.
- FLWR iskazi.
- Uslovni iskazi.
- Kvantifikovani iskazi.
- Iskazi koji u sebi uključuju korisnički definisane funkcije.

Najpoznatiji sistemi za upravljanje izvornim XML bazama podataka su:

- *eXist*, open source sistem za upravljanje izvornim XML bazama podataka, koji jednostavno može biti integrisan u druge aplikacije koje koriste i obraduju XML. Baza podataka je potpuno napisana u Javi.
- *Berkeley DB XML*.
- *Oracle XML DB*.
- *MarkLogic Server*, izvorna XML baza podataka koja koristi XQuery.

U ovom radu, za dobijanje podataka iz leksičkih resursa, korišćen je eXist sistem za upravljanje izvornim XML bazama.

3. Postojeće metode klasifikacije

Nakon prikupljanja tekstualnih dokumenata u korpus i prikaza dokumenata na način razumljiv za računar, sledeći korak koji treba napraviti je izgradnja samog modela klasifikacije ili klasifikatora na osnovu skupa za učenje. Najznačajnije metode klasifikacije zasnovane na mašinskom učenju su [75]:

- Metode zasnovane na drvetima odlučivanja (eng. decision trees).
- Metode zasnovane na pravilima (eng. rule based).
- Metode zasnovane na rastojanju (najbliži sused, eng. nearest neighbour).
- Statistički zasnovane metode.
- Metode zasnovane na neuronskim mrežama (eng. neural networks).
- Metode zasnovane na podržavajućim vektorima (eng. support vector machines).

3.1 Metode zasnovane na drvetima odlučivanja

Drveta odlučivanja su moćne i popularne tehnike modeliranja podataka koje se koriste, između ostalog, i u cilju rešavanja problema klasifikacije. Privlačnost ove metode leži u činjenici da metoda nudi model podataka u obliku "čitljivom" i razumljivom za čoveka, odnosno u obliku pravila. Ta pravila se mogu lako direktno interpretirati običnim jezikom a mogu se i koristiti u nekom od jezika za rad sa bazama podataka (SQL). Tako se određeni primeri iz baze mogu izdvojiti korišćenjem pravila generisanih drvetima odlučivanja.

Drvo odlučivanja ima stabloliku strukturu. Razlikuju se tri vrste čvorova (čvor je generičko ime za element drveta) [75]:

- *Koren* (eng. root node), koji nema ulazne i ima nula ili više izlaznih grana (putanja).

- *Unutrašnji čvor* (eng. internal node), koji ima jednu ulaznu i dve ili više izlaznih grana. Korenu i svim unutrašnjim čvorovima pridruženi su uslovi testiranja atributa (eng. attribute test condition). Na osnovu toga da li je neki uslov pridružen čvoru ispunjen ili ne, vrši se razdvajanje primera sa različitim karakteristikama, odnosno različitim vrednostima atributa. Zbog toga se ovi čvorovi nazivaju još i čvorovi odluke (eng. decision nodes).

- *List* (eng. leaf, terminal node), koji ima jednu ulaznu i nijednu izlaznu granu. Listovi predstavljaju sva moguća rešenja problema. To su čvorovi koji definišu klasu kojoj pripadaju primeri koji zadovoljavaju sve uslove na grani drveta kojoj pripada taj list.

Drvo odlučivanja može da se koristi za klasifikaciju primera (slogova, instanci) na sledeći način: krene se od korena drveta i ide se po onim granama koje primer sa svojim vrednostima atributa zadovoljava. Tako se ide sve do krajnjeg čvora ili lista koji predstavlja jednu od postojećih klasa problema kojoj se primer pridružuje.

Osnovni preduslovi za korišćenje ove metode su:

- Primeri moraju biti opisani preko konačnog broja atributa.

- Odgovori na pitanja koja se javljaju u rešavanju problema su iz nekog predefinisanog skupa pitanja.

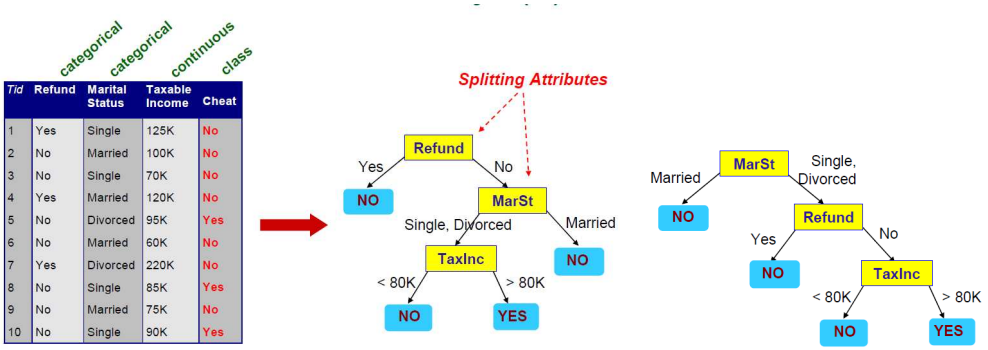
- Klase kojima primeri pripadaju moraju biti unapred definisane i mora ih biti konačan broj.

- Za izgradnju klasifikacionog modela potreban je značajan broj primera za učenje (barem nekoliko stotina).

Za određeni problem klasifikacije i zadati skup primera za učenje postoji eksponencijalno mnogo potencijalnih drveta koji mogu biti konstruisani (videti sliku 3.1). Razvijen je čitav niz različitih algoritama za konstruisanje drveta odlučivanja. Jedan takav je i Hantov algoritam, koji je osnova mnogim drugim algoritmima kao na primer ID3, C4.5 i CART [75].

3.1.1 Hantov algoritam

Kod Hantovog algoritma drvo odlučivanja se gradi rekurzivno. Neka je D_t skup primera za učenje koji su pridruženi čvoru t, a $y = \{y_1, ..., y_c\}$ skup svih mogućih klasa. Algoritam se sastoji od rekurzivne primene sledeća dva koraka:



Slika 3.1: Primeri drveta odlučivanja.

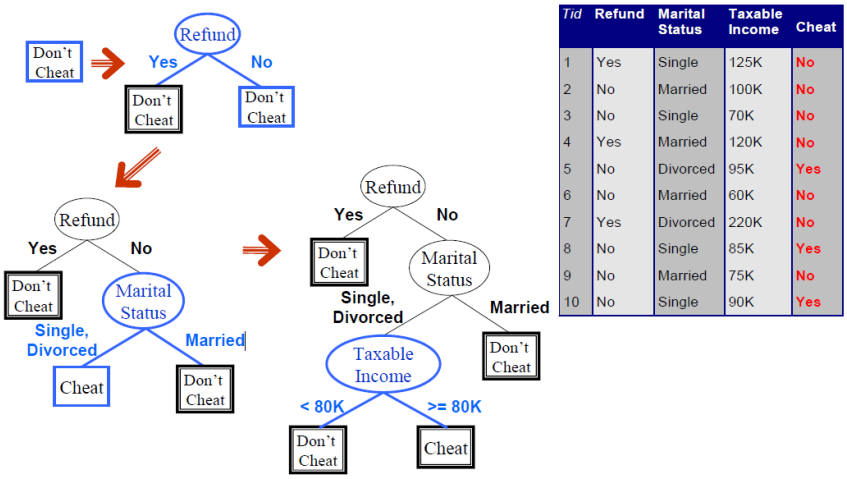
1. Ako svi primeri iz skupa D_t pripadaju jednoj klasi y_t (kaže se da je tada čvor t čist), onda je čvor t zapravo list drveta i on sve primere koji su njemu pridruženi dodeljuje klasi y_t.

2. Ako primeri iz skupa D_t pripadaju različitim klasama, bira se neki test atribut, odnosno uslov čvora kojim će biti podeljen skup D_t na više manjih podskupova. Za svaki mogući rezultat ispunjenja uslova čvora postoji po jedno dete čvor. Primeri skupa D_t se pojedinačno dodeljuju odgovarajućim dete-čvorovima u zavisnosti od toga kako ispunjavaju uslov. Nakon raspoređivanja svih primera, oba koraka se rekurzivno primenjuju nad svakim novim čvorom.

Ilustracija Hantovog algoritma prikazana je na slici 3.2. Može se primetiti da postoji više načina na koje se skup primera za učenje može podeliti na podskupove. To zavisi od izbora test atributa i uslova testiranja. Dakle, prva odluka koju treba doneti je kako podeliti primere za učenje odnosno koji atribut izabrati za test atribut, kako navesti uslove testiranja za atribute različitih tipova i kako odrediti najbolju podelu (ako postoji više njih). Takođe se iz algoritma može primetiti da se on izvršava sve dok svi čvorovi ne budu čisti. To je u praktičnoj upotrebi neefikasno sa stanovišta izračunljivosti. Zbog toga je potrebno doneti odluku o kriterijumu zaustavljanja algoritma [75].

3.1.2 Uslovi testiranja za atribute

Broj različitih ishoda ispunjenja uslova čvora jednak je broju izlaznih grana iz tog čvora. Sam uslov testiranja atributa koji je pridružen čvoru zavisi od tipa test atributa (imenski, redni, neprekidni) i od broja načina za deobu,



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

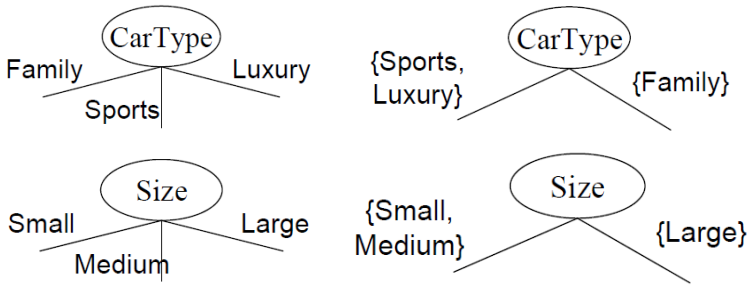
Slika 3.2: Ilustracija Hantovog algoritma.

odnosno broja mogućih ishoda uslova (podela na dve i više grana). Primeri podele zasnovane na imenskim i rednim atributima kada se podela vrši na više grana i kada se vrši na dve grane prikazani su na slici 3.3. Kod podele na više grana koristi se toliko grana koliko ima različitih mogućih vrednosti atributa dok kod binarne podele vrednosti se dele u dva podskupa. U slučaju podele zasnovane na neprekidnim atributima, diskretizacijom se formiraju redni kategorički atributi. To može biti izvršeno statički (diskretizacija se vrši jednom, na početku) i dinamički (opsezi mogu da se odrede podelom na jednake intervale, jednaku frekvenciju, percentile, klastere i drugo). Kod binarne podele, vrši se izbor neke fiksne vrednosti v i onda se podela vrši u zavisnosti od toga da li je vrednost atributa manja ili veća od v. Primer podele zasnovane na neprekidnim atributima prikazan je na slici 3.4.

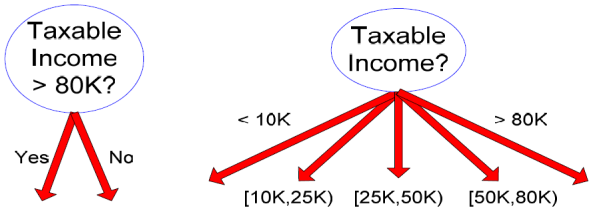
Pitanje koje se postavlja je kako odrediti najbolju podelu odnosno koji uslov testiranja atributa daje najbolje rezultate.

3.1.3 Pristup pohlepe

Pristup pohlepe koristi načelo da podela koja daje najhomogenije podskupove (podskupove u kojima većina primera pripada istoj klasi – nizak nivo nečistoće) jeste najbolja podela. Na primer, ukoliko u nekom skupu od 10 primera, 5 pripada jednoj a 5 drugoj klasi, smatra se da je taj skup nehomogen i da ima visok nivo nečistoće. Ako u slučaju istog skupa, 9 primera pripada jednoj a samo jedan primer pripada drugoj klasi, onda je skup homogen odnosno ima



Slika 3.3: Podela zasnovana na imenskim i rednim atributima: na više grana i binarna podela.



Slika 3.4: Podela zasnovana na neprekidnim atributima: na više grana i binarna podela.

nizak nivo nečistoće.

Najbolja podela može da se dobije sprovođenjem sledeća dva koraka:

- Posmatranom čvoru kome treba odrediti uslov testiranja, vrši se merenje trenutne nečistoće skupa primera koji mu pripadaju.
- Za različite moguće uslove pravi se podela i računa se nečistoća u novodobijenim čvorovima. Stepen nečistoće nakon podele mora biti manji nego što je bio pre podele. Računa se razlika odnosno dobit u čistoći na sledeći način:

$$Dobit = I(otac_{cvor}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

pri čemu je I mera nečistoće, v_j je dete čvor, k je broj novodobijenih dete-čvorova, $N(v_j)$ broj primera koje pripadaju dete-čvoru v_j, N je ukupan broj primera koji pripadaju otac-čvoru (čvoru gde se pravi podela).

Ona podela koja donosi najveću dobit je najbolja podela. Postoje različite mere nečistoće čvora a neke od njih su: Ginijev indeks, Entropija i Greška u klasifikaciji.

Ginijev indeks

Za dati čvor drveta t, Ginijev indeks (Corrado Gini, italijanski statističar) se računa po sledećoj formuli:

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

pri čemu je $p(j|t)$ relativna frekvencija klase j u čvoru t. Maksimum ove mere postiže se kada su primeri ravnomerno raspoređeni u svim klasama (najveći stepen nečistoće) a minimum kada sve instance pripadaju jednoj klasi (najmanji stepen nečistoće). Primeri računanja Ginijevog indeksa prikazani su na slici 3.5.

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

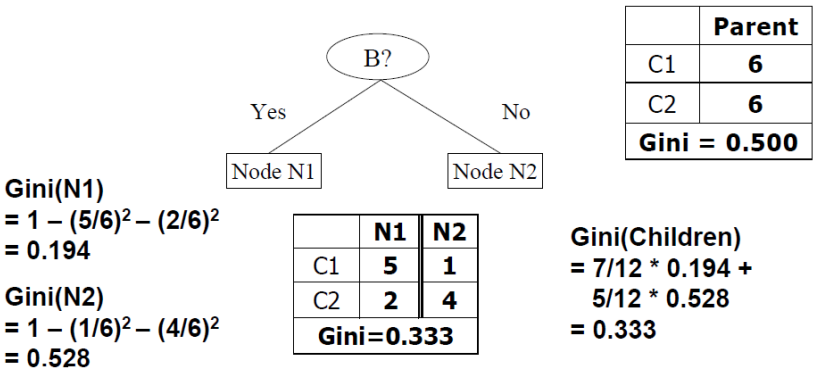
C1	3
C2	3
Gini=0.500	

Slika 3.5: Primer izračunavanja Ginijevog indeksa.

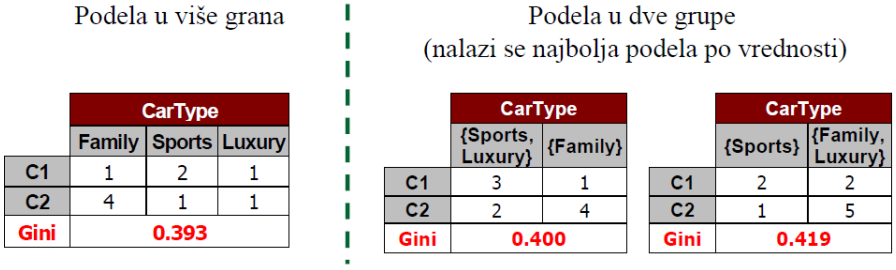
Primer izračunavanja Ginijevog indeksa kod podele po binarnim atributima prikazan je na slici 3.6. Skup se deli u dve particije pričemu su poželjnije veće i čistije particije. Primer izračunavanja Ginijevog indeksa kod podele po kategoričkim atributima prikazan je na slici 3.7. U ovom slučaju, u donošenju odluke koristi se matrica brojanja.

Kod podele po neprekidnim atributima, koriste se binarne pitalice zasnovane na jednoj vrednosti. Ta jedna vrednost može biti izabrana na onoliko načina koliko ima različitih vrednosti. Svakoj vrednosti po kojoj se vrši deljenje pridružuje se matrica brojanja (u svakoj od particija prebrojavaju se klase, vrednost atributa je manja ili veća od vrednosti v). Jednostavan način za izbor najboljeg v je da se za svako v skenira baza podataka kako bi se dobila matrica brojeva i izračunao Ginijev indeks. Ovo zahteva ponavljanje posla i neefikasno je sa stanovišta izračunljivosti.

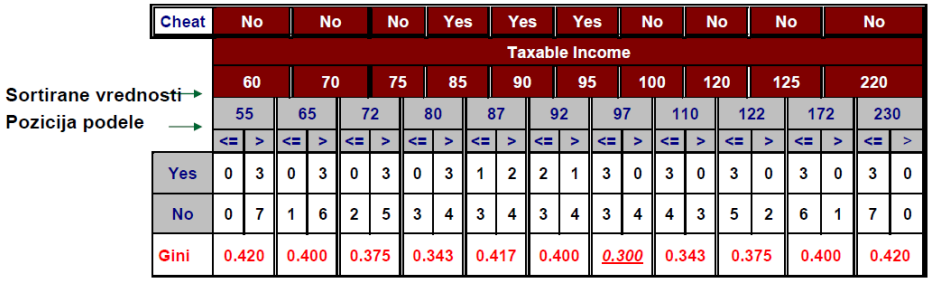
Za efikasno izračunavanje, za svaki atribut se vrši sortiranje po vrednostima atributa. Dobijene vrednosti se zatim linearno skeniraju uz ažuriranje matrice brojanja i izračunavanje Ginijevog indeksa. Bira se pozicija za podelu sa najmanjim Ginijevim indeksom (videti sliku 3.8) [75].



Slika 3.6: Primer izračunavanja Ginijevog indeksa – podela po binarnim atributima.



Slika 3.7: Primer izračunavanja Ginijevog indeksa – podela po kategoričkim atributima.



Slika 3.8: Primer izračunavanja Ginijevog indeksa – podela po neprekidnim atributima.

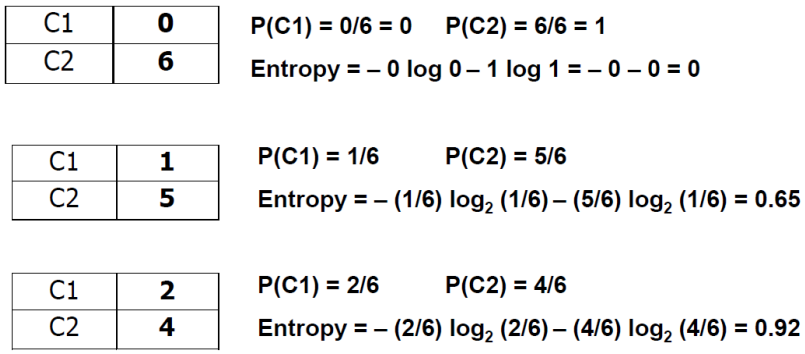
Alternativni kriterijumi podele

Još jedna poznata mera homogenosti čvora je entropija. Entropija se u datom čvoru t računa po sledećoj formuli:

$$Entropy(t) = -\sum_j p(j|t) log_2 p(j|t)$$

pričemu je $p(j|t)$ relativna frekvencija klase j u čvoru t. Maksimum ove mere se dobija kada su primeri ravnomerno raspoređeni u svim klasama a minimum se dobija kada svi primeri pripadaju jednoj klasi.

Primer računanja entropije dat je na slici 3.9.



Slika 3.9: Primer izračunavanja entropije.

Izračunavanja zasnovana na entropiji i Ginijevom indeksu su slična. Obe mere teže da biraju atribute koji imaju veliki broj različitih vrednosti, što znači da teže ka kreiranju podela sa velikim brojem novih i čistijih čvorova, što nekad nije željeni rezultat.

Osim Ginijevog indeksa i entropije, popularna je i mera greške pri izračunavanju:

$$Error(t) = 1 - \max_i P(i|t)$$

Primer izračunavanja ove mere dat je slikom 3.10.

Na slici 3.11 je prikazano poređenje među merama nečistoće kod binarne klasifikacije.

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Error = 1 – max (0, 1) = 1 – 1 = 0

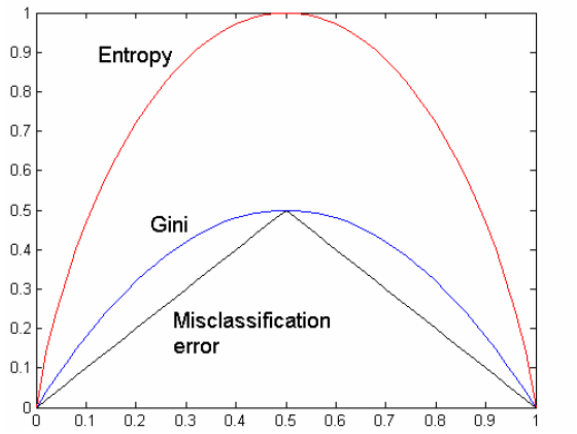
C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6
Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6
Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Slika 3.10: Primer greške pri izračunavanju.



Slika 3.11: Poređenje mera nečistoće za problem binarne klasifikacije.

Mera kvaliteta podele

Kada se čvor p deli na k delova (dete-čvorova) kvalitet podele se računa kao:

$$I_{split} = \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j)$$

pri čemu je v_j dete čvor, k je broj novodobijenih dete-čvorova, $N(v_j)$ je broj primera koji pripadaju dete-čvoru v_j a N je ukupan broj primera koji pripadaju otac-čvoru. $I(v_j)$ je neka od mera nečistoće čvora (Ginijev indeks, Entropija ili Greška pri izračunavanju).

Dobit podele se u tom slučaju računa kao:

$$GAIN_{split} = I(otac_{cvor}) - I_{split}$$

Podela se bira tako da se dobija najveća redukcija (maksimizira se dobit $GAIN$). Ovaj način se koristi u ID3 i C4.5 algoritmu.

Odnos dobiti se koristi za određivanje valjanosti podele. U C4.5 se kao kriterijum valjanosti koristi:

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO}$$

$$SplitINFO = -\sum_{j=1}^{k} \frac{N(v_j)}{N} log \frac{N(v_j)}{N}$$

pri čemu je v_j dete čvor, k je broj novodobijenih dete-čvorova, $N(v_j)$ je broj primera koji pripadaju dete-čvoru v_j a N je ukupan broj primera koji pripadaju otac-čvoru.

3.1.4 Karakteristike drveta odlučivanja

Neke važnije prednosti drveta odlučivanja u odnosu na druge tehnike su [75]:

- Tehnika za konstruisanje drveta odlučivanja nije skupa, što omogućava brzo dobijanje modela čak i kada je skup za učenje veoma velik.
- Klasifikovanje nepoznatog materijala u izgrađenom modelu je brzo i efikasno.
- Tehnika je laka za interpretaciju, posebno za drveta male veličine.
- Prilično je otporna na postojanje šuma.
- Tačnost ove tehnike je uporediva sa ostalim tehnikama klasifikacije za jednostavne tipove podataka.

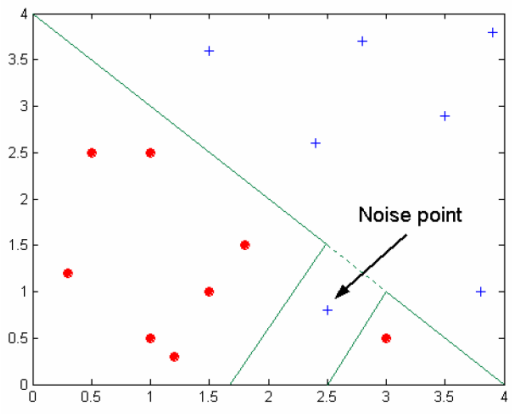
Praktični problemi pri klasifikaciji su [75]:

- Previše i premalo prilagođen model (eng. overfitting, underfitting).
- Nedostajuće vrednosti.
- Cena klasifikacije.

Dobar klasifikacioni model mora raditi dobro, odnosno malo grešiti i na skupu za učenje i na skupu za testiranje. Greška pri klasifikaciji se deli u dva tipa: greška u fazi učenja (broj loše klasifikovanih primera iz skupa za učenje) i greške u generalizaciji (očekivana greška modela nad nepoznatim primerima). Model koji se dobro ponaša na skupu za učenje a loše na skupu za testiranje, odnosno, model kod koga je greška pri učenju mala a pri testiranju značajno veća, naziva se *previše prilagođen* model (eng. overfitting). Sa druge strane, ako je model suviše jednostavan tako da su greške i pri učenju i pri generalizaciji (uopštavanju, testiranju) visoke, onda je reč o *premalo prilagođenom* modelu (eng. underfitting).

Postoji više razloga zbog kojih može doći do previše prilagođenog modela: prisustvo šuma, nepostojanje reprezentativnih primera i drugo.

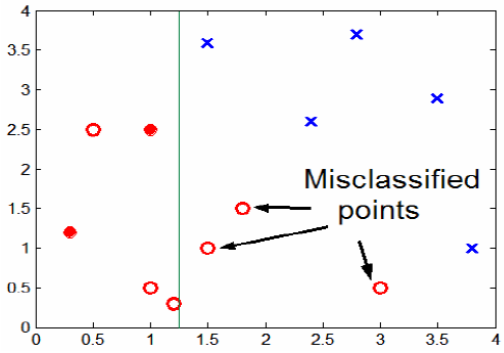
Kako šum utiče na preveliku prilagođenost modela najbolje se može ilustrovati slikom 3.12. Na ovoj slici, plusevi i kružići predstavljaju primere dve različite klase. Plave linije predstavljaju dva moguća razgraničenja klasa kao rezultat klasifikacije. Previše prilagođen model, kod koga se važnost daje šumu, predstavljen je iskrivljenim linijama.



Slika 3.12: Prevelika prilagođenost zbog šuma.

Modeli koji formiraju kriterijum klasifikacije na osnovu malog broja reprezentativnih primera za učenje, su takođe podložni prevelikoj prilagođenosti. Drvo koje se gradi da bi postiglo malu grešku nad primerima za čenje, prilagođava se većini primera. U ovom slučaju to nisu reprezentativni primeri, što znači da će se na skupu za testiranje ovaj model dosta lošije pokazati. Na primer, na slici 3.13 nedostatak tačaka u donjoj polovini dijagrama onemogućava korektno predviđnje oznaka klasa u tom delu. U procesu konstruisanja klasifikatora se koriste primeri za učenje koji su irelelvantni za klasifikaciju u

tom delu dijagrama.



Slika 3.13: Preprilagođenost zbog nepostojanja reprezentativnih primera.

Može se zaključiti da drvo ne treba da bude ni previše složeno i duboko (zbog prevelike prilagođenosti) ni previše jenostavno i plitko (zbog premale prilagođenosti). Pri izgradnji drveta, ako se samo minimalizuje greška pri učenju, ne dobije se korektna procena ponašanja drveta u slučaju pojave prethodno nepoznatih primera. Potrebno je naći korektnu procenu greške generalizacije. Minimalizujući nju, dolazi se do optimalnijih modela. Problem je lako uočljiv: Kako proceniti grešku generalizacije odnosno grešku modela pri klasifikaciji nepoznatih primera, kada pri učenju skup za testiranje nije unapred poznat? Ipak postoje metode koje se bave procenom greške generalizacije.

3.1.5 Procena greške u generalizaciji

Neka je T drvo, t čvor, N broj listova u drvetu T, $e(t)$ broj pogrešno klasifikovanih primera u čvoru t, $e(T)$ ukupan broj pogrešno klasifikovanih primera za drvo T, $e^{'}(T)$ procena greške generalizacije za drvo T.

Optimistički pristup se zasniva na tome da su podaci za učenje reprezentativni. To znači da greška pri učenju treba da bude dobar pokazatelj i za grešku generalizacije, što daje: $e^{'}(t) = e(t)$, odnosno $e^{'}(T) = e(T)$.

Pesimistički pristup se zasniva na tome da se uz grešku svakog čvora dodaje i takozvana kazna za kompleksnost drveta. Formula se može prikazati na sledeći način:

$$e_g^{'}(T) = \frac{\sum_{i=1}^{k}[e(t_i) + \omega(t_i)]}{\sum_{i=1}^{k} n(t_i)} = \frac{e(T) + \omega(T)}{N_t}$$

pri čemu je $\omega(t_i)$ kazna za kompleksnost drveta u čvoru t_i, N_t broj primera u skupu za učenje a $n(t_i)$ broj primera za učenje u čvoru t_i.

Primer 3.1 *Neka je kazna za kompleksnost drveta za svaki čvor, konstantno jednaka* 0.5. *Neka drvo ima 4 lista, i neka je 6 primera loše klasifikovano od ukupno 24 primera za učenje. Pesimističkim pristupom, greška generalizacije može da se izračuna na sledeći način:*

$$e_g = \frac{6 + 4 * 0.5}{24} = \frac{1}{3}$$

Ovaj pristup dobro je kombinovati sa principom štedljivosti koji se naziva još i Okamov (eng. Occam) žilet. Ovaj pristup uveden je Okamovom definicijom koja glasi:

Definicija 3.1 *Od dva modela sa istom greškom generalizacije treba izabrati onaj koji je jednostavniji.*

Drugi način da se složenost modela uključi u samu procenu modela je princip najmanje dužine opisa (eng. Minimum Description Length, MDL). Ovaj princip se takođe oslanja na to da manje kompleksni modeli znače i bolji modeli, ali se kompleksnost modela računa na poseban način. Treba uzeti onaj model čija je cena da se enkodira sam model i cena da se enkodiraju svi nekorektno klasifikovani primeri skupa za učenje, najmanja. Zapravo, potrebno je minimalizovati zbir te dve cene. Važi [75]:

$$Cena(model, podaci) = Cena(model) + Cena(podaci|model)$$

pri čemu je $Cena$ broj bitova potreban za enkodiranje. $Cena(model)$ sadrži cenu enkodiranja modela a $Cena(podaci|model)$ sadrži cenu enkodiranja pogrešno označenih primera pri klasifikaciji. Traži se najmanje skup model, odnosno model koji minimizuje ukupnu cenu.

Cela priča o proceni greške generalizacije (uopštavanja) ima za cilj nalaženje optimalnih modela i izbegavanje prevelike prilagođenosti. Pored ovoga, moguće je poštovati još dve strategije za izbegavanje potencijalno prevelikog prilagođavanja, u kontekstu drveta odlučivanja. To su pre-potkresivanje (eng. pre-pruning) i post-potkresivanje (eng. post-pruning) [75].

3.1.6 Pre-potkresivanje i post-potkresivanje

Pre-potkresivanje se zasniva na tome da se algoritam kreiranja drveta zaustavlja pre nego što drvo naraste do maksimalne veličine. To se može postići

na više načina a najčešće korišćeni su: postavljanje praga za dozvoljeni koeficijent nečistoće listova i postavljanje praga za minimalan broj primera koji mora da postoji u nekom čvoru da bi se nastavilo sa pravljenjem podela.

Kada se drvo izgradi do kraja, onda na scenu stupa post-potkresivanje. Seku se čvorovi odozdo na više tamo gde je moguće da se celo poddrvo zameni jednim listom. Uglavnom, ako se greška generalizacije poboljša posle otsecanja poddrveta i postavljanja lista, onda to otsecanje ima smisla i nastavlja se dalje proces post-potkresivanja. Ovaj pristup ima prednost u odnosu na pre-potkresivanje, jer se radi na već potpuno izgrađenom drvetu, i ne postoji opasnost da se preuranjeno otseče neko poddrvo. Međutim, to što radi na potpuno izgrađenom drvetu, ima svoju lošu stranu. Naime, potrebno je dosta vremena za izgradnju celog drveta, a dobar deo tog rada kasnije se odbacuje otsecanjem.

3.1.7 Postojeći sistemi i domeni primene

Postoji puno klasifikacionih sistema koji su zasnovani na drvetima odlučivanja. Neki od njih su slobodno dostupni a neki su komercijalni. Od slobodno dostupnih najpoznatiji su sledeći sistemi:

- **C4.5**[1]**.** Ovo je jedan od najpopularnijih sistema za klasifikaciju zasnovanih na drvetima odlučivanja. Njegov autor je Ross Quinlan [61] . Predstavlja unapređenje *ID3* algoritma (od istog autora) koji može da radi samo sa kategoričkim podacima. Za razliku od njega, C4.5 može da radi i sa numeričkim podacima. Dobro podržava rad sa nedostajućim vrednostima i ima mogućnost post-potkresivanja. Otvorenog je koda. Namenjen je za rad na Unix (Linux) platformi. Poznata implementacija ovog algoritma u java programskom jeziku je *J48* klasifikator koji je deo poznatog *WEKA* alata za istraživanje podataka. Ima veoma širok domen primene.

- **YaDT: Yet another Decision Tree builder**[2]**.** Ovaj sistem predstavlja novu implementaciju C4.5 drveta odlučivanja. Može se koristiti na Windows (Visual Studio) i Linux (gcc) sistemima. Poboljšan je mogućnošću paralelnog izvršavanja na višejezgarnim računarima (samo za Linux). Za prikaz drveta na izlazu koristi se PMML (engl. Predictive Model Markup Language). Implementiran je u C++ programskom jeziku.

[1] http://www.rulequest.com/Personal/
[2] www.di.unipi.it/∼ruggieri/software

- **OC1**[3] (eng. Oblique Classifier 1) [53]. Ovaj sistem je prilagođen za rad sa numeričkim vrednostima. Drvo odlučivanja se generiše pomoću linearne kombinacije jednog ili više atributa u svakom unutrašnjem čvoru. Sadrži veliki broj funkcija za podršku fleksibilnom ekperimentisanju sa različitim tipovima podataka. Između ostalog, ovaj sistem obezbeđuje unakrsnu validaciju eksperimenata i grafički prikaz skupa podataka i drveta odlučivanja. Napisan je u ANSI C programskom jeziku. Ima primenu u različitim domenima kao što su astronomija, dijagnostikovanje kancera i drugo.

- **CHAID**[4] (eng. CHi-squared Automatic Interaction Detection). Ovo je jedan od najstarijih sistema za klasifikaciju zasnovanih na ne-binarnim drvetima odlučivanja koje je 1980. godine predložio Kass [37]. Zasnovan je na prilično jednostavnom algoritmu dobro prilagođenom analizi većih skupova podataka.

Neki od najpoznatijih komercijalnih sistema su:

- **AC2**[5]**.** Ovaj klasifikacioni sistem daje mogućnost korisniku da kreira i manipuliše drvetima dobijenim od podataka koji mogu biti strukturirani ili ravni (eng. flat) (najčešće u formatu matrice). Sistem ima dobro dizajniran i atraktivan interfejs koji omogućava jaku interakciju sa korisnikom. Drvo odlučivanja se prikazuje grafički i omogućava korisniku da lako pregleda čvorove, menja ih i lako vrši testiranja. Implementiran je u C++ programskom jeziku. Ima primenu u raznim domenima kao što su bankarstvo, marketing, analiza rizika, kontrola kvaliteta, medicinska dijagnostika i epidemiologija, analiza i tipologija stanovanja i drugo. *Alice d'Isoft 6.0* je unapređena verzija ovog sistema namenjena za poslovnu upotrebu.

- **C5.0/See5**[6]**.** Ovaj alat predstavlja naslednika popularnog *C4.5* algoritma. U poređenju sa njim, proizvodi značajno manje drvo odlučivanja i ima manju vremensku složenost a tačnost klasifkacije je zanemarljivo veća. Pogodan je za rad sa velikim bazama podataka i prilagođen je radu na računarima sa do osam jezgara i jednim ili više procesora. Klasifikatori konstruisani ovim sistemom su dostupni u vidu C izvornog koda tako da se mogu lako integrisati u neki drugi sistem. Dostupan je za Windows Xp/Vista/7/8 (See5) i Linux (C5.0) operativne sisteme.

[3] http://www.cbcb.umd.edu/∼salzberg/announce-oc1.html
[4] http://www.statsoft.com/textbook/chaid-analysis/
[5] http://www.alice-soft.com/
[6] http://www.rulequest.com/

- **CART 5.0 decision-tree software**[7]**.** CART je robustan sistem jednostavan za korišćenje namenjen radu sa velikim i kompleksnim bazama podataka. Dostupan je za MVS (eng. Multiple Virtual Storage), SMS (eng. Systems Management Server), Windows i Linux operativne sisteme. Ima primenu u domenima kao što su profilisanje kupaca, ciljana direktna pošta, otkrivanje prevara kreditnim karticama i upravljanje kreditnim rizikom.

- **XpertRule Miner**[8]**.** Ima veoma dobro razvijen grafički korisnički interfejs. Vizuelizacija dobijenih rezultata je podržana kroz 2D i 3D slike. Podržava višeslojnu klijent-server arhitekturu. Dozvoljena je mogućnost pristupa bilo kojoj bazi podataka uz podršku ODBC (eng. Open Database Connectivity) konekcije. Dostupan je za Windows operativni sistem.

3.2 Klasifikatori zasnovani na pravilima

Klasifikatori zasnovani na pravilima predstavljaju tehniku za klasifikaciju primera korišćenjem skupova pravila oblika "*if...then...*" koji se mogu prikazati na sledeći način [75]:

$$Pravilo : (USLOV) \longrightarrow y$$

pri čemu je leva strana pravila takozvani *preduslov pravila* i sastoji se od konjukcije atributa, a desna strana pravila se naziva *posledica pravila* i sadrži informaciju o klasi. Formalni opšti oblik preduslova pravila može se dati u sledećem obliku:

$$USLOV = (A_1 \ op \ v_1) \wedge ... \wedge (A_k \ op \ v_k)$$

pri čemu je $(A_k \ op \ v_k)$ par atribut-vrednost, a *op* je logički operator koji pripada skupu $\{=, \neq, <, \leq, \geq, >\}$. Svaki par atribut-vrednost $(A_k \ op \ v_k)$ naziva se konjunkt.

Primer 3.2 *Primer pravila za klasifikaciju je:*

$$(Toplokrvna = da) \wedge (Nosi \ jaja = Da) \longrightarrow Ptice$$

$$(Oporezivi \ prihod < 50K) \wedge (Vraca = Da) \longrightarrow Izbegava = Ne$$

Kaže se da pravilo r *pokriva* (*obuhvata*) primer x, ako vrednosti atributa primera x zadovoljavaju preduslov pravila r. Takođe, kaže se da je pravilo *pokrenuto* ili *izazvano* svaki put kada pokrije neki primer.

[7] http://www.salford-systems.com/
[8] http://www.attar.com/

3.2.1 Kvalitet klasifikacionog pravila

Kvalitet klasifikacionog pravila može da se proceni na osnovu dve karakteristike: *preciznost* i *odziv* (pokrivenost). *Preciznost* predstavlja procenat broja primera koji zadovoljavaju desnu stranu pravila od svih primera koji zadovoljavaju levu stranu pravila a *odziv* predstavlja procenat broja primera koji zadovoljavaju levu stranu pravila. Formalno, neka je D skup primera a $r : A \longrightarrow y$ pravilo. Tada se preciznost i odziv mogu definisati na sledeći način [75]:

$$Preciznost = \frac{|A|}{|D|}$$

$$Odziv = \frac{|A \cap y|}{|A|}$$

pri čemu je $|A|$ broj primera koji ispunjavaju preduslov pravila r, $|A \cap y|$ broj primera koji ispunjavaju preduslov pravila r i čija je klasa y a $|D|$ je broj primera u skupu primera D.

Način rada klasifikatora zasnovanog na pravilima biće ilustrovan sledećim primerom:

Primer 3.3 *Dat je skup pravila:*

$$R1 : (Radja = ne) \wedge (Leti = da) \longrightarrow Ptice$$

$$R2 : (Radja = ne) \wedge (Zivi\ u\ vodi = da) \longrightarrow Ribe$$

$$R3 : (Radja = da) \wedge (Toplokrvna = da) \longrightarrow Sisari$$

$$R4 : (Radja = ne) \wedge (Leti = ne) \longrightarrow Reptili$$

$$R5 : (Zivi\ u\ vodi = ponekad) \longrightarrow Vodozemci$$

Korišćenjem ovako definisanog modela, izvršiti klasifikaciju sledećeg skupa primera za testiranje:

Ime	Toplokrvnost	Rađa	Leti	Živi u vodi	Klasa
Lemur	da	da	ne	ne	?
Kornjača	ne	ne	ne	ponekad	?
Morski pas	ne	da	ne	da	?

Rešenje: *Lemura* pokriva pravilo $R3$ i prema tome ovaj skup pravila, odnosno ovaj klasifikator će ga označiti kao *sisara*. *Kornjaču* pokrivaju pravila $R4$ i $R5$, pri čemu ova dva pravila klasifikuju primere koje pokrivaju različite klase, što znači da se ne može odrediti klasa za *kornjaču*. Problem je i primer *morski pas* kojeg ne pokriva ni jedno pravilo.

3.2.2 Karakteristike klasifikatora zasnovanih na pravilima

Klasifikatori zasnovani na pravilima karakterišu se sa dve osnovne osobine [75]:

- *Uzajamno isključiva pravila.* Pravila nekog skupa su uzajamno isključiva ako ne postoje dva pravila iz skupa pravila koja pokrivaju jedan isti primer. Dakle, svaki primer pokriva najviše jedno pravilo.
- *Iscrpno pokrivanje.* Svaki primer pokriven je bar jednim pravilom.

Ove dve karakteristike zajedno daju da svaki primer mora biti pokriven tačno jednim pravilom. Tako nešto nije baš često pa zato postoje tehnike za popravljanje, tako da skup pravila ipak ispuni ove dve osobine. Primer koji je malopre naveden sadrži skup pravila koji niti ima uzajamno isključiva pravila niti iscrpno pokrivanje, što znači da je to prilično loš skup pravila.

Popravljanje skupa pravila

Ako skup nije *iscrpan*, onda je pametno dodati jedno novo predefinisano pravilo, koje će pokrivati sve one primere koji do sada nisu pokriveni.

Ako pravila nisu uzajamno isključiva, postoje dva načina da se popravi skup pravila. Prvi način je da se skup pravila uredi (sortira, rangira) prema prioritetu pravila tako da će primer biti pokriven prvim pravilom najvišeg prioriteta čiji preduslov ispunjava. Uređen skup pravila poznat je i kao lista odlučivanja. Drugi način je poznat pod nazivom *glasački sistem.* Primer se propušta kroz ceo skup pravila i izdvajaju se ona pravila koja je primer izazvao. Vrši se prebrojavanje takozvanih *glasova* za svaku potencijalnu klasu. Klasa sa najvećim brojem glasova pobeđuje. Dodatno, ako je neko pravilo značajnije od nekog drugog, može se otežati tako što se na primer množi sa koeficijentom njegove preciznosti. Prednosti ove stratgije su što pravi manje grešaka i skup pravila ne mora da se čuva u sortiranom obliku, međutim problem je što se svaki primer propušta kroz ceo skup pravila, što može da bude poprilično skupo.

Prilikom određivanja uređenja skupa pravila postoje dva osnovna pristupa: *uređenje zasnovano na pravilima* (pojedinačna pravila se rangiraju prema njihovom kvalitetu), *uređenje zasnovano na klasama* (pravila koja pripadaju istoj klasi se grupišu jedno do drugog).

3.2.3 Kreiranje klasifikatora

Kreirati klasifikator ovde konkretno znači odrediti skup pravila koji prepoznaje ključne veze između vrednosti atributa i klasa objekata. Postoje dve široke grupe metoda za dobijanje skupa pravila [75]:

- *Direktne metode*, gde se pravila izdvajaju direktno iz podataka (neki od primera takvih klasifikatora su RIPPER, CN2, 1R)
- *Indirektne metode*, gde se pravila izdvajaju iz nekih drugih klasifikacionih modela izgrađenih nad istim skupom primera za učenje, kao na primer iz drveta odlučivanja, neuronskih mreža i drugo (primer takvog klasifikatora je C4.5)

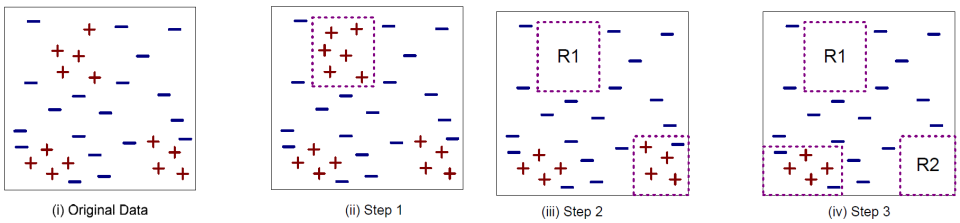
3.2.4 Direktna metoda: sekvencijalno pokrivanje

Sekvencijalno pokrivanje je algoritam koji se veoma često koristi za kreiranje klasifikacionog modela. Kod ovog algoritma izdvajaju se pravila za svaku klasu redom. Sam redosled klasa zavisi od raznih faktora kao na primer, frekvencija pojavljivanja klase u skupu za učenje ili cena pogrešne klasifikacije za klasu koja je bitna po nekom kriterijumu, i tako dalje. Koraci u izvršavanju algoritma su sledeći:

1. Počinje se od praznog skupa pravila.
2. Za klasu koja je na redu, izdvajaju se svi pozitivni i negativni primeri. Pozitivini primeri su primeri iz te klase a negativni su svi ostali primeri.
3. Skup pravila se proširuje korišćenjem funkcije *Learn-One-Rule*. Naime, za izdvojene skupove pozitivnih i negativnih primera, traži se pravilo r koje će pokriti većinu pozitivnih i nijedan negativan primer (bar ne značajan broj negativnih primera). To se radi korišćenjem pomenute *Learn-One-Rule* funkcije. Kada se pronađe takvo pravilo, prelazi se na sledeći korak.
4. Svi primeri koji su pokriveni novodobijenim pravilom iz prethodnog koraka, izbacuju se iz skupa za učenje a pravilo r se dodaje skupu pravila.
5. Algoritam sada nastavlja izvršavanje od koraka 2 rekurzivno, do dostizanja kriterijuma zaustavljanja.

Primer sekvencijalnog pokrivanja prikazan je na slici 3.14.

Iz algoritma se može zaključiti da su osnovne osobine sekvencijalnog pokrivanja: porast pravila, eliminacija primera, provera pravila, kriterijum zaustavljanja, potkresivanje (skupa) pravila.



Slika 3.14: Primeri sekvencijalnog pokrivanja.

Porast pravila

Postoje dve strategije za porast pravila:

- *Opšte* $\longrightarrow$ *Specifično*: Počinje se od praznog pravila $r: \longrightarrow y$ gde je preduslov prazan. Ovo pravilo pokriva sve primere iz skupa za čenje. Preduslovu se dodaju novi poduslovi koji su međusobno u konjuktivnoj vezi. Dodaje se prvo onaj poduslov koji ima najveću čistoću, zato se ovaj pristup zove još i pristup pohlepe. Postupak se ponavlja sve dok dobijeno pravilo ne zadovolji kriterijum zaustavljanja (na primer, kada se dodavanjem novog poduslova ne dobija bolji kvalitet pravila).

- *Specifično* $\longrightarrow$ *Opšte*: Na slučajan način se bira neki pozitivan primer za predodređenu klasu. Prema tom primeru se u potpunosti prilagođava jedno pravilo i ono predstavlja inicijalno pravilo ovog pristupa. To pravilo se zatim profinjuje: jedan po jedan poduslov se izbacuje sve dok novodobijeno pravilo ne počne da pokriva negativne primere. Uvek se izbacuju prvo oni poduslovi koji time daju veće pokrivanje pozitivnih primera.

Pomenuta funkcija *Learn-One-Rule* iz algoritma, može da implementira jedno od ova dva pristupa. Pravilo koje je dobijeno, može na kraju biti potkresano. Naime, ako se uklanjanjem konjunkta iz pravila dobija smanjena greška generalizacije, onda se taj konjunkt svakako izbacuje.

Postoje razne direktne metode koje se uglavnom razlikuju po načinu na koji mere kvalitet pravila (CN2 entropijom a RIPPER FOIL-om) i načinom na koji vrše potkresivanje pravila. RIPPER algoritam koristi posebnu meru za potkresivanje $v = \frac{p-n}{p+n}$, gde su p broj pozitivnih a n broj negativnih primera.

Eliminacija primera

Eliminacija primera se vrši kako naredno pravilo ne bi bilo identično prethodnom pravilu. Pozitivni primeri se eliminišu kako bi se obezbedilo da je sledeće pravilo različito od postojećih a negativni primeri se eliminišu da bi bilo onemogućeno smanjenje preciznosti pravila.

Provera pravila

Provera kvaliteta pravila može se postići jednom od metrika:

$$Preciznost = \frac{n_c}{n}$$

$$Laplas = \frac{n_c + 1}{n + k}$$

$$M - procenat = \frac{n_c + kp}{n + k}$$

pri čemu je n broj primera pokriven pravilom, n_c je broj pozitivnih primera pokrivenih pravilom, k je broj klasa i p je ranija verovatnoća za pozitivne klase.

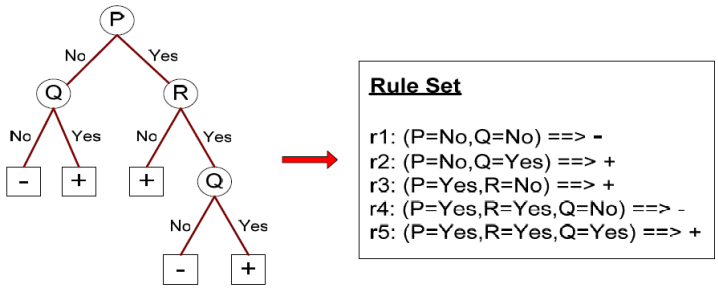
Kriterijum zaustavljanja i potkresivanje pravila

Kriterijum zaustavljanja može biti sledeći: izračuna se dobit dodavanja novog pravila i ako dobit nije značajna, to novo pravilo se odbacuje. Potkresivanje pravila je slično potkresivanju drveta: uklanja se jedan od konjukata u pravilu i ako se greška smanjuje, isključuje se konjukt iz pravila.

3.2.5 Indirektne metode

Kao što je ranije rečeno, kod indirektnih metoda pravila se izdvajaju iz nekih drugih klasifikacionih modela izgrađenih nad istim primerima za učenje, kao na primer drveta odlučivanja. Naime, putanja od korena do lista drveta odlučivanja može da se izrazi kao jedno pravilo. Ako se tako izraze sve putanje, dobija se skup pravila koji potpuno odgovara modelu drveta, a koji je pritom *iscrpan* i pravila su mu međusobno *isključiva*. Primer je prikazan na slici 3.15

Jedan od najpoznatijih algoritama indirektne metode je *C4.5rules* algoritam.



Slika 3.15: Ilustracija indirektne metode.

C4.5rules

Algoritam se sastoji iz sledećih koraka:

1. Izdvojiti pravila iz nepotkresanog drveta odlučivanja.
2. Za svako pravilo $r : A \longrightarrow y$:
   (a) Razmotriti sva alternativna pravila $r' : A' \longrightarrow y$ gde je A' dobijeno uklanjanjem jednog konjukta iz A.
   (b) Porediti pesimistički nivo greške za r u odnosu na sve r'-ove. Izdvojiti pravilo $r*$ sa najmanjom pesimističkom greškom, ukoliko uopšte to alternativno pravilo ima pesimističku grešku manju nego kod originalnog pravila.
   (c) Izvršiti potkresivanje originalnog pravila.
   (d) Ponavljati postupak sve dok se pesimistička greška pri uopštavanju poboljšava.

Kada se algoritam završi i dobije se skup pravila onda je potrebno izvršiti nad njim određeno uređenje. Ovaj algoritam koristi uređenje zasnovano na klasama. Dakle, kreira podskupove pravila gde sva pravila jednog podskupa imaju istu posledicu (desnu stranu, klasu). Zatim se računa dužina opisa za svaki podskup, po sledećoj formuli [75]:

$$DuzinaOpisa = L(greske) + g * L(model)$$

pri čemu je g parametar koji ima predefinisanu vrednost 0.5 i koji se štimuje u zivisnosti od prisustva redudantnih atributa u skupu pravila (što je veći broj redundantnih atributa to je g manje).

Kada se izračuna dužina opisa za svaki podskup, onda se pravi raspored podskupova, odnosno kreira se takozvana *lista odlučivanja* gde se podskupovi pravila sortiraju rastuće prema svojoj dužini opisa. Očekuje se da podskup koji ima najmanju dužinu opisa nudi najbolja pravila.

3.2.6 Prednosti klasifikatora zasnovanih na pravilima

Klasifikacione modele zasnovane na pravilima karakterišu sledeće osobine:

- Izražajna moć ista je kao i kod drveta odlučivanja.
- Jednostavna interpretacija, odnosno iz njih se lako mogu uočiti i vizualizovati veze između atributa.
- Jednostavno formiranje.
- Mogu brzo da klasifikuju nove primere.
- Performanse su uporedive sa drvetima odlučivanja.

3.2.7 Postojeći sistemi i domeni primene

Od postojećih klasifikacionih sistema zasnovanih na pravilima najpoznatiji su sledeći:

- **SuperQuery**[9]**.** Ovaj alat se može koristiti za Microsoft Access baze, Excel tabele i mnoge druge baze podataka. On omogućava automatsko generisanje grafikona i prikaz različitih statistika bez prethodnog poznavanja SQL-a ili nekog drugog jezika za rad sa bazama. Ovaj alat se koristi za poslovnu upotrebu: u prodaji, finansijama, računovodstvu i drugim domenima.
- **WizWhy**[10]**.** Ovo je alat za automatsko otkrivanje pravila u podacima na osnovu kojih se može vršiti dalja analiza podataka. Može se koristiti za Microsoft SQL i Oracle baze, za Access baze i može se povezati na bilo koju drugu bazu korišćenjem ODBC konekcije. Može čitati bilo koju ASCII datoteku. Dostupan je za Windows operativni sistem. Koristi se u raznim domenima kao na primer, kreditni rizik u bankarstvu i poslovanju, rizik osiguranja, direktni marketing, zadržavanje kupaca, detekcija prevara, dijagnostikovanje u medicini i drugo.

[9] http://www.azmy.com/
[10] http://www.wizsoft.com/

- **XpertRule Miner**[11]**.** Ovaj alat obezbeđuje kompletnu analizu podataka kroz grafički korisnički interfejs zasnovan na ikonicama. Rezultati se vizuelno mogu prikazati preko 2D i 3D grafova i slika. Pomoću ODDBC konekcije može da pristupi bilo kojoj bazi podataka. Dostupan je za Windows operativni sistem.

- **CBA**[12]**.** Ovo je jedan od najpoznatijih slobodno dostupnih sistema ove vrste. Ovaj sistem iz datih relacionih podataka proizvodi pravila pridruživanja na osnovu kojih generiše klasifikatore. Postoji i složeniji alat *DM-II system* koji sadrži *CBA* i joňeke alate za dodatnu analizu i obradu pravila pridruživanja. Domeni primene su marketing, finansije, bankarstvo, proizvodnja, telekomunikacije i drugi.

3.3 Metoda k-najbližih suseda

Za razliku od metoda zasnovanih na pravilima, postoje i metode zasnovane na instancama (primerima). Za razliku od nekih drugih metoda kod kojih se vrši konstruisanje opšteg, generalnog oblika ciljne funkcije, kod metode na osnovu primera postupak generalizacije odlaže se do trenutka predstavljanja novog primera kojeg treba klasifikovati. Takve metode nazivaju se *lenje metode* (eng. lazy methods). Jedan primer ovakvih metoda je *učenje napamet*. Kod ove metode klasifikacija se sprovodi samo ako se atribiti primera za testiranje u potpunosti poklope sa atributima nekog primera za učenje. Primer za testiranje se tada klasifikuje isto kao primer za učenje.

Druga, manje lenja metoda, je metoda k-najbližih suseda (eng. k-nearest neighbour, k-NN). Za realizaciju ove metode potreban je skup sačuvanih primera za učenje, metrika za izračunavanje rastojanja između primera i vrednost k koja predstavlja broj najbližih suseda koji se uzimaju u razmatranje. Radi klasifikacije novog primera za testiranje, izračunava se rastojanje od tog primera do ostalih primera za učenje i određuje se k najbližih suseda, odnosno k primera za učenje sa najmanjim rastojanjem od tog primera. Koristeći oznake klasa tih najbližih suseda, određuje se oznaka klase primera za testiranje (na primer, uzima se oznaka većine).

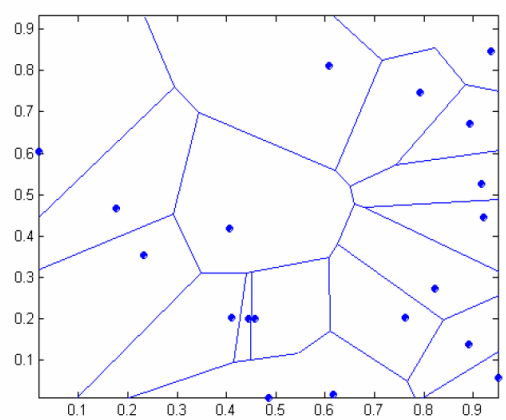
U k-NN algoritmu, postoji pretpostavka da se svi primeri mogu predstaviti tačkama u n-dimenzionalnom prostoru R^n. Primer x je opisan vektorom atributa $\langle a_1(x), ..., a_n(x)\rangle$, gde $a_i(x)$ označava vrednost i-tog atributa primera x. Udaljenost između primera x_i i x_j definiše se Euklidovim rastojanjem na sledeći način:

[11] http://www.attar.com/
[12] http://www.comp.nus.edu.sg/∼dm2

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n}(a_r(x_i) - a_r(x_j))^2}$$

Ciljna funkcija u k-NN algoritmu, može imati diskretne ili neprekidne vrednosti. Ciljna funkcija sa diskretnim vrednostima definiše se kao $f : R^n \longrightarrow V$, gde je $V = \{v_1, ...v_s\}$ konačan skup od s oznaka klasa. Vrednost koju vraća ova ciljna funkcija je zapravo oznaka klase kojoj pripada većina od k najbližih suseda. Ako se za vrednost k izabere broj 1 tada će vrednost ciljne funkcije biti oznaka klase pridružena najbližem primeru za učenje od primera kojeg treba klasifikovati.

S obzirom da k-NN algoritam ne definiše opštu ciljnu funkciju za celi prostor primera, postavlja se pitanje kako bi takva funkcija mogla izgledati. Odgovor na to pitanje bio bi rezultat ispitivanja prikazanog u radu [52], u kojem bi se naučenim k-NN klasifikatorom mogao klasifikovati svaki mogući primer iz prostora primera. Za $k = 1$ prostor primera podeljen je na poligone čije stranice određuju granice područja unutar kojih su svi mogući primeri za testiranje bliži primeru za učenje unutar tog regiona nego bilo kom drugom primeru za učenje. To je ilustrovano slikom 3.16 gde tačke predstavljaju primere za učenje i unutar svakog poligona postoji samo po jedan primer. Svaki od poligona određuje područje u kom će se novi primer za testiranje klasifikovati u klasu kojoj pripada primer za učenje u tom poligonu. Ovaj dijagram zove se još Voronoi dijagram.



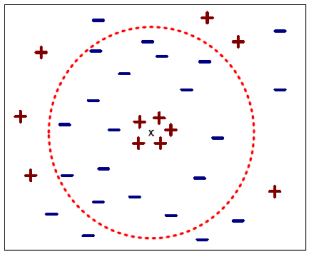
Slika 3.16: Voronoi dijagram za 1-nn i dvodimnenzionalne podatke.

Algoritam k-NN se lako može prilagoditi radu sa neprekidnim vrednostima. Umesto prebrojavanja koliko od k najbližih suseda pripada kojoj klasi,

računa se njihova srednja vrednost.

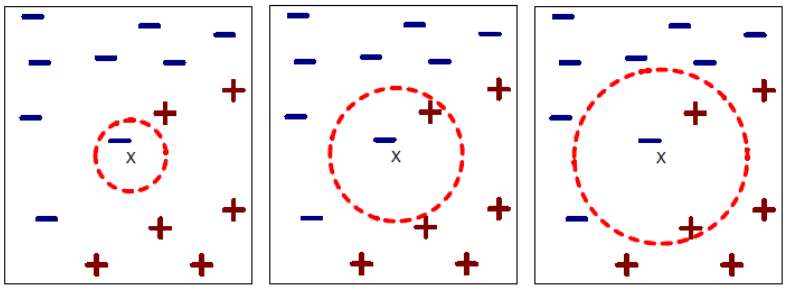
U cilju što efikasnijeg procesa klasifikacije k-NN algoritmom, potrebno je izvršiti dobar izbor vrednosti za k (videti sliku 3.17) [75]:

- Ako je k jako malo, klasifikacija je osetljiva na šum.
- Ako je k jako veliko, susedi mogu da uključe tačke iz drugih klasa.



Slika 3.17: Klasifikacija kNN metodom za veliko k.

U praksi je k najčešće neparan broj da bi se izbegli slučajevi kada jednak broj suseda upada u više klasa (videti sliku 3.18 za $k = 2$). Sa ove slike jasno se može videti da se primer za testiranje, u zavisnosti od odabranog k, može različito klasifikovati: zajedno sa minusevima za $k = 1$ a sa plusevima za $k = 3$ [75].



Slika 3.18: Klasifikacija k-NN metodom za $k = 1$, $k = 2$ i $k = 3$.

3.3.1 Algoritam k-najbližih suseda

Algoritam k-najbližih suseda izračunava udaljenost (ili razliku) između svakog primera za testiranje $z = (x', y')$ i svih primera za učenje $(x, y) \in D$ kako bi

odredio najbliže susede svakom primeru za testiranje. Ovakvo izračunavanje može biti skupo ako je skup primera za učenje velik, pa se koriste efikasne metode indeksiranja kako bi se redukovala potrebna količina izračunavanja.

Koraci u primeni k-NN algoritma mogu se definisati na sledeći način [75]:

- Neka je k broj najbližih suseda i D skup svih primera za učenje.
- Za svaki primer za testiranje $z = (x', y')$ sa nepoznatom klasifikacijom, uraditi sledeće:
  - Izračunati rastojanje $d(x', x)$ između z i svakog primera $(x, y) \in D$.
  - Selektovati podskup $D_z \subseteq D$ od k najbližih primera za učenje primeru z.
  - Vratiti oznaku klase kao rezultat izračunavanja funkcije:

$$y' = arg\max_{v \in V} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$$

    pri čemu je v oznaka klase, y_i oznaka klase nekog od najbližih suseda, a $I()$ je funkcija koja vraća vrednost 1 ako je vrednost njenog argumenta tačno (eng. true), a 0 ako je netačno (eng. false).

3.3.2 Uvođenje težinskih faktora

Budući da algoritam radi na intuitivnoj pretpostavci da su najbliži primeri potencijalno slični, poboljšanje algoritma koje se samo nameće je uvođenje težinskih faktora. Za svaki od k suseda, uvodi se težinski faktor w_i koji iznosi inverznu vrednost kvadrata njegove udaljenosti od primera za testiranje $z = (x', y')$. Tada se vrednost ciljne funkcije određuje formulom:

$$y' = arg\max_{v \in V} \sum_{(x_i, y_i) \in D_z} w_i I(v = y_i)$$

gde je

$$w_i = \frac{1}{d(x', x_i)^2}$$

Ako se primer za testiranje poklapa sa primerom za učenje, odnosno udaljenost između njih je jednaka nuli, primer za testiranje se klasifikuje isto kao taj primer za učenje. Ukoliko se više primera preklapaju, uzima se

klasifikacija većine primera. Modifikacija za slučaj neprekidne ciljne funkcije definiše se na sledeći način:

$$y' = \frac{\sum_{i=1}^{k} w_i y'_i}{\sum_{i=1}^{k} w_i}$$

gde je težinski faktor w_i definisan kao i za prethodni slučaj.

Zbog uvođenja težisnkih faktora, udaljeni primeri će imati vrlo malo uticaja na klasifikaciju pa nestaje potreba za ograničavanjem broja suseda koji se uzimaju u razmatranje. Metoda kod koje se svi primeri za učenje uzimaju u obzir pri procesu klasifikacije naziva se *globalnom*, a ako se njihov broj ograničava, metoda se naziva *lokalnom*. Globalna metoda primenjena na neprekidne vrednosti naziva se još i *Shepardova metoda*.

3.3.3 Karakteristike k-NN algoritma

Osnovne karakteristike k-NN algoritma su sledeće:

- *Induktivna pristrasnost*. Pretpostavlja se da je klasifikacija upita slična klasifikaciji primera u blizini.
- *Kletva dimenzionalnosti* (eng. curse of dimensionality). Udaljenost se računa na osnovu svih atributa pa je algoritam osetljiv na sve atribute bez obzira na njihov broj i značaj za ciljnu funkciju. Jedno od rešenja bi moglo biti množenje atributa sa težinskim faktorima kako bi se smanjio uticaj nevažnih atributa. Kod drastičnijeg pristupa, faktori mogu imati vrednost 0 čime se mogu u potpunosti ukloniti nevažni atributi.
- *Složenost izračunavanja*, zbog čega je potrebno ostvariti efikasno indeksiranje memorije.

3.3.4 Postojeći sistemi i domeni primene

Neki od poznatijih klasifikacionih sistemima zasnovanih na matodi k najbližih suseda su **PEBLS** [10], koji je napisan i C programskom jeziku i ima primenu u predviđanju sekundarne strukture proteina i identifikaciji DNK promoter sekvenci, i **Evidential distance-based classifier** [13] [15].

[13] https://www.hds.utc.fr/∼tdenoeux

3.4 Bajesova metoda

U mnogim situacijama, između skupa atributa kojima je određen primer za testiranje i oznake klase, odnos je ne-deterministički. Drugim rečima, oznaka klase primera za testiranje ne može da se predvidi sa sigurnošću, čak i ako je njen skup atributa isti kao skup atributa nekog primera za učenje. To se dešava zbog postojanja i nekih drugih faktora koji utiču na klasifikaciju a nisu uključeni u analizu. Na primer, može se posmatrati zadatak predviđanja da li je neka osoba u riziku da oboli od bolesti srca ili ne, na osnovu njenog načina ishrane i učestalosti vežbanja. Iako osobe koje se zdravo hrane i redovno vežbaju imaju manje šanse da obole od bolesti srca, ipak to može da im se desi zbog nekih drugih faktora kao na primer nasledni faktor, pušenje, korišćenje alkohola i drugo.

Bajesova metoda predstavlja verovatnosno okruženje za rešavanje problema klasifikacije. Metoda se zasniva na *Bajesovskoj teoremi* koja kombinuje prethodno znanje o klasama sa novim znanjima dobijenim iz podataka za učenje. Bajesova teorema glasi:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

pri čemu su $P(Y|X)$ i $P(X|Y)$ uslovne verovatnoće.

Ilustracija Bajesove teoreme data je sledećim primerom:

Primer 3.4 *Poznate su sledeće činjenice:*

- *Doktor zna da meningitis u* 50% *slučajeva prouzrokuje ukočenost vrata.*
- *Prethodna (poznata) verovatnoća da bilo koji pacijent ima meningitis je* $\frac{1}{50000}$.
- *Prethodna verovatnoća da bilo koji pacijent ima ukočen vrat je* $\frac{1}{20}$.

Ako pacijent ima ukočen vrat, koja je verovatnoća da ima i meningitis?

Rešenje:

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{\frac{1}{2}\frac{1}{50000}}{\frac{1}{20}} = 0.0002$$

pri čemu M označava da pacijent ima meningitis a S označava da pacijent ima ukočen vrat.

Neka je X neki primer sa skupom atributa $(A_1, ..., A_n)$ a C oznaka klase. Cilj je predvideti klasu C kojoj pripada primer X. $P(C|X)$ je verovatnoća da se dogodi C ako se dogodilo X, odnosno, verovatnoća da primer X pripada

klasi C. Ideja je da se maksimizuje $P(C|X)$. Postavlja se pitanje: da li može da se proceni $P(C|X)$ direktno na osnovu podataka za učenje?

Postupak se sastoji iz dva koraka:

1. Vrši se izračunavanje posledične (eng. posterior) vrednosti $P(C|A_1A_2...A_n)$ za svaku vrednost C koristeći Bajesovu teoremu.

2. Vrši se izbor vrednosti C koja maksimizuje vrednost $P(C|A_1A_2...A_n)$

Drugi korak je ekvivalentan sa tim da se nađe C koje maksimizuje $P(A_1A_2...A_n|C)P(C)$. Pitanje koje se nameće je kako izračunati $P(A_1A_2...A_n|C)$?

3.4.1 Naivni Bajesov klasifikator

Prema Naivnoj Bajesovoj metodi, pretpostavlja se uslovna nezavisnost između atributa A_i za $i = 1, ..., n$. Ova pretpostavka o uslovnoj nezavisnosti može se formalno iskazati na sledeći način:

$$P(A_1A_2...A_n|C) = P(A_1|C)P(A_2|C)...P(A_n|C)$$

pri čemu se pojedinačne verovatnoće $P(A_i|C)$ za $i = 1, ..., n$ mogu relativno lako izračunati iz podataka za učenje.

Dakle, Naivni Bajesov klasifikator klasifikuje nepoznate primere na osnovu Bajesove teoreme i podataka za učenje ali uz jednu značajnu pretpostavku a to je *nezavisnost atributa*.

Naivni Bajesov klasifikator označava primer zadat atributima $(A_1, ..., A_n)$ klasom C_* ako važi:

$$\max_{C_j}\{P(C_j)\prod_{k=1}^{n}P(A_k|C_j)\} = P(C^*)\prod_{k=1}^{n}P(A_k|C^*)$$

Za diskretne atribute verovatnoća $P(A_i|C)$ se generalno može lako izračunati. Postavlja se pitanje, šta ako je atribut A_i neprekidan?

Procena uslovnih verovatnoća za neprekidne atribute

Postoje dva načina da se odrede uslovne verovatnoće za neprekidne atribute u Naivnom Bajesovom klasifikatoru [75]:

1. Prvi način je diskretizacija u grupe. Svaka neprekidna vrednost atributa može da se zameni sa odgovarajućim diskretnim intervalima. Dobije

se jedan redni atribut po grupi. Procena greške zavisi od same strategije diskretizacije ali i od broja intervala diskretizacije. Treba odabrati pravi broj intervala ali i same intervale, odnosno granice. Ako je broj intervala premali onda je moguće da će neki intervali sadržati primere za učenje koji pripadaju različitim klasama, što ne daje dobre rezultate. S druge strane, ako je broj intervala prevelik onda svaki interval sadrži suviše mali broj primera za učenje da bi se obezbedila pouzdana procena uslovne verovatnoće.

2. Drugi način se zasniva na pretpostavci da neprekidni atribut ima određenu raspodelu verovatnoća. Često se u ove svrhe koristi *Gausova normalna raspodela*. Ova raspodela karakteriše se sa dva parametra: *sredina* μ i *varijansa* ili *disperzija* σ^2. Kada je raspodela verovatnoća poznata, ona se može koristiti za procenu uslovnih verovatnoća za atribut A_i i za svaku klasu C_j:

$$P(A_i|C_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

pri čemu srednja vrednost μ_{ij} predstavlja matematičko očekivanje ili srednju vrednost vrednosti A_i za sve podatke za učenje koji pripadaju klasi C_j, a varijansa ili disperzija σ_{ij}^2 predstavlja matematičko očekivanje odstupanja vrednosti A_i od srednje vrednosti, za sve primere za učenje koji pripadaju klasi C_j.

Često se dešava da je neka od uslovnih verovatnoća jednaka nuli, što onda celokupnu procenu anulira i naravno narušava ceo algoritam, odnosno onemogućava klasifikaciju u tim slučajevima. Ovaj problem može biti prevaziđen korišćenjem *m-procene*. Procena verovatnoća problematičnih primera vrši se pomoću sledeće formule:

$$P(A_i|C_j) = \frac{n_c + mp}{n + m}$$

pri čemu je n_c broj primera za učenje koji pripadaju klasi C_j i čiji atributi ispunjavaju uslov A_i, n je ukupan broj primera klase C_j a m i p su parametri (unapred fiksirane realne vrednosti).

3.4.2 Karakteristike Naivnog Bajesovog klasifikatora

Naivni Bajesov klasifikator generalno ima sledeće osobine [75]:

- Klasifikator je tolerantan na izolovani šum jer ga on jednostavno "izravna" srednjim vrednostima pri određivanju uslovnih verovatnoća. Takođe, prevazilazi problem nedostajućih vrednosti tako što ignoriše takve primere u procesu izračunavanja procene verovatnoće.

- Klasifikator dosta dobro toleriše irelevantne atribute. Ako je A_i irelevantan atribut onda $P(A_i|C)$ ima gotovo uniformnu raspodelu i gotovo da nema uticaja na izračunavanje celokupne verovatnoće.

- Visoko korelisani atributi mogu značajno da ugroze kvalitet ovog klasifikatora jer za njih pretpostavka o nezavisnosti nikako ne važi. Primenom ovog algoritma dobijaju se loše procene.

3.4.3 Postojeći sistemi i domeni primene

Neki od poznatijih klasifikacionih sistema zasnovanih na Bajesovoj metodi su:

- **Bayes Server**[14]. Ovo je sistem koji sadrži korisnički interfejs i API (eng. Application Programming Interface) za izgradnju i vizuelizaciju modela, za učenje modela iz podataka, za crtanje grafika, predviđanje vremenskih serija i drugo. Iako ima podršku za kontinualne podatke, diskretizacija podataka je poželjna. Tačnost klasifikacije se može prikazati pomoću matrice konfuzije i grafika.

- **Bayes Classifier**[15]**.** Dostupan za Linux i Windows operativni sistem. Napisan u C programskom jeziku.

- **Bayesian belief network software (Win95/98/NT/2000)**[16]**.** Ovaj alat je slobodno dostupan i uključuje *BN PowerConstructor*, efikasan sistem za učenje parametara iz podataka, *BN PowerPredictor*, sistem za klasfikaciju i modelovanje podataka i *Data PreProcessor*, alat koji se koristi za preprocesiranje podataka za učenje.

- **NBC**. Ovo je jednostavan klasifikacioni sistem napisan u awk programskom jeziku.

[14] http://www.bayesserver.com/
[15] http://www.borgelt.net//software.html#bayes
[16] http://www.cs.ualberta.ca/∼jcheng/bnsoft.htm

3.5 Skriveni Markovljevi modeli

Iako su prvi put predstavljene i proučavane krajem šezdesetih i početkom sedamdesetih godina prošlog veka, statističke metode *Markovljevi modeli* i *Skriveni Markovljevi modeli* su tek u poslednjoj deceniji postale veoma popularne. Postoje dva ključna razloga za to: prvi leži u činjenici da su ovako definisani modeli vrlo bogate strukture i predstavljaju dobru osnovu za opis velikog broja različitih problema, a drugi razlog je taj što je njihova struktura veoma složena pa je neophodno imati jaku računarsku podršku u implementaciji ovih modela.

Markovljevi modeli prvog reda predstavljaju stohastičke modele kojima se vrši predviđanje stanja nekog sistema u periodu koji sledi na osnovu poznavanja njegovog trenutnog stanja i verovatnoća promena tog stanja u periodu predviđanja. U slučaju modela reda k, predviđanje stanja sistema se vrši na osnovu njegovog trenutnog stanja, *k-1* prethodnih stanja i verovatnoća promena tog stanja u periodu predviđanja. Ovde će biti opisan samo model prvog reda.

Neka je dat sistem takav da se u svakom trenutku može naći u jednom od N unapred poznatih mogućih stanja $1, 2, ..., N$. Neka su u jednakim vremenskim intervalima moguće promene stanja sistema u skladu sa unapred definisanim verovatnoćama i neka važi sledeća relacija:

$$P[q_t = j | q_{t-1} = i, q_{t-2} = k, ...] = P[q_t = j | q_{t-1} = i]$$

pri čemu je q_t stanje u kome se sistem nalazi u trenutku t. Ovo zapravo znači da to u koje će stanje sistem preći u sledećem trenutku zavisi isključivo od njegovog trenutnog stanja, a ne od stanja koja su mu prethodila. Da bi sistem mogao da se u potputnosti opiše, neophodno je poznavanje verovatnoće prelaska sistema iz jednog stanja u drugo. Ove verovatnoće a_{ij} mogu da se definišu na sledeći način:

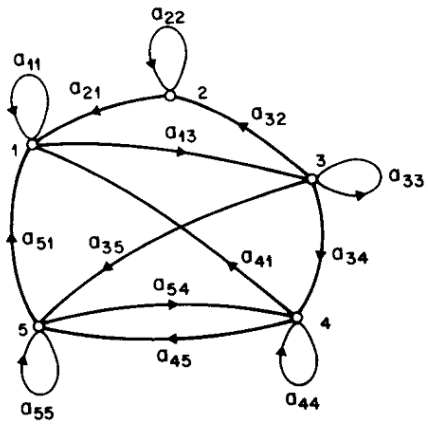
$$a_{ij} = P[q_t = j | q_{t-1} = i], 1 \leq i, j \leq N$$

Parametri a_{ij} se nazivaju *verovatnoće prelaza* i moraju da zadovolje sledeće uslove:

$$a_{ij} \geq 0$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

Za ovako definisan sistem kaže se da je merljiv Markovljevim modelom. Primer Markovljevog modela sa 5 stanja i izabranim verovatnoćama prelaza stanja prikazan je na slici 3.19.



Slika 3.19: Diskretni Markovljev model sa 5 stanja i izabranim verovatnoćama prelaza stanja.

Model koji je prikazan na slici 3.19, u kome se podrazumeva da je svako stanje sistema vidljivo (merljivo), prilično je restriktivan da bi bio primenljiv na čitave klase problema koji su od interesa. Proširenje ovakvog koncepta vodi ka skrivenim Markovljevim modelima. Kod ovakvih modela razlikuju se *ispoljeni* (uočeni, engl. observed) delovi problema i *skriveni* delovi problema. Neka je ispoljena sekvenca označena sa $X = X_1, X_2, ..., X_L$ a sevenca skrivenih stanja sa $\pi = \pi_1, \pi_2, ..., \pi_L$. U modelu prikazanom na slici 3.19 bilo je jasno koje je stanje odgovorno za svaki deo uočene sekvence. Kod skrivenih Markovljevih modela postoji više stanja koja mogu da budu odgovorna za svaki deo uočene sekvence. Imajući ispoljenu sekvencu na raspolaganju nije jasna slika o tome u kom se stanju sistem trenutno nalazi, i to predstavlja skriveni deo problema. Kao i u Markovljevom modelu, mogu se razlikovati sledeći parametri: *verovatnoće prelaza*

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k)$$

koje predstavljaju verovatnoću prelaza iz stanja k u stanje l, i *emisione verovatnoće*:

$$e_k(b) = P(X_i = b | \pi_i = k)$$

koje predstavljaju verovatnoću emitovanja simbola b u stanju k.

Dobar primer ilustracije ovog modela jeste primer nepoštene kockarnice.

Primer 3.5 ***Model nepoštene kockarnice:*** *Kockar baca kockice igrajući neku igru. Poznato je da raspolaže kockicom koja nije fer (ne daje sve brojeve*

sa jednakom verovatnoćom). Na raspolaganju je sekvenca brojeva dobijenih nakon što je kockar n puta bacio kockicu:

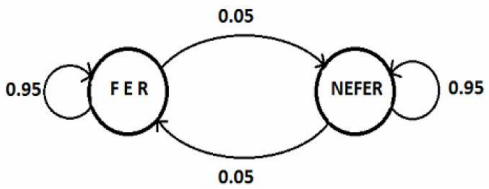
$$X = X_1X_2X_3...X_n = 1\ 2\ 3\ 5\ 1\ 2\ 1\ 5\ 6\ 4$$

Formirati Markovljev model koji će da opiše prikazanu ispoljenu sekvencu.

Rešenje: Prvo pitanje koje se postavlja je šta treba da budu stanja ovog modela i koliko treba da ih bude. Ne zna se sa koliko kockica kockar raspolaže niti koju kockicu u datom trenutku koristi.

Najjednostavniji način je da se pretpostavi da kockar ima samo jednu nefer kockicu. U ovom slučaju Markovljev model nije skriven (u svakom trenutku poznata je kockica koja se baca) i jedino pitanje je koliko je kockica nefer, odnosno sa kolikim verovatnoćama se dobija svaki od brojeva od 1 do 6. To znači da ovaj model ima 5 nepoznatih parametara. To su verovatnoće da će pasti brojevi od 1 do 5 (verovatnoća da će pasti broj 6 može da se dobije kada se od broja 1 oduzme zbir verovatnoća za brojeve od 1 do 5).

Druga mogućnost je da se formira skriveni Markovljev model. Pretpostavka je da postoje dve kockice, fer i nefer, koje predstavljaju dva različita stanja. U ovakvom modelu problem je odrediti verovatnoće prelaza iz jednog u drugo stanje (prelaz sa jedne na drugu kockicu) kao i emisione verovatnoće (verovatnoće pojave brojeva od 1 do 6 u slučaju izbora fer i nefer kockice). To znači da u ovom slučaju postoji 12 nepoznatih parametara. Na slici 3.20 je dat šematski prikaz ovog modela sa verovatnoćama prelaza i emisionim verovatnoćama za fer i nefer kockicu.



P(1|F) = 1/6	P(1|N) = 1/10
P(2|F) = 1/6	P(2|N) = 1/10
P(3|F) = 1/6	P(3|N) = 1/10
P(4|F) = 1/6	P(4|N) = 1/10
P(5|F) = 1/6	P(5|N) = 1/10
P(6|F) = 1/6	**P(6|N) = 1/2**

Slika 3.20: Primer nepoštene kockarnice: verovatnoće prelaza i emisione verovatnoće.

Pitanje koje se postavlja je koji od ova dva modela više odgovara realnoj situaciji, odnosno dobijenoj ispoljenoj sekvenci. Model sa većim brojem parametara ima veći stepen slobode pa je jasno da drugi model može bolje da opiše ispoljenu sekvencu. Ipak, mora da se uzme u obzir da je ovaj model složeniji pa se pri izboru modela mora naći neka odgovarajuća mera između složenosti i tačnosti modela.

3.5.1 Elementi skrivenih Markovljevih modela

Skriveni Markovljev model može da se definiše kao uređena petorka:

$$(\sum, Q, \pi, A, E)$$

koju čine sledeći elementi:

- $\sum$ je azbuka ispoljenih ili emisionih simbola. Emisioni simboli odgovaraju fizičkom merenju modela. Za primer nepoštene kockarnice to su brojevi od 1 do 6.
- Q je skup stanja. Iako su stanja skrivena, u najvećem broju praktičnih problema postoji jasna njihova fizička interpretacija i značenje. Na primer, u slučaju nepoštene kockarnice stanje odgovara konkretnoj kockici koja je izabrana. U opštem slučaju, stanja se definišu tako da se u svako stanje može stići iz bilo kog drugog stanja. Obično se sa $Q = \{1, 2, ..., N\}$ označava skup svih mogućih stanja, a sa q_t stanje modela u trenutku t.
- π je vektor početnih verovatnoća. U primeru nepoštene kockarnice, ako kockar sa jednakim verovatnoćama bira kojom će kockicom početi igru, taj vektor je $(\frac{1}{2}, \frac{1}{2})$. Zbir ovih verovatnoća je 1.
- $A = \{a_{ij}\}$ je matrica verovatnoća prelaska. Verovatnoća prelaska iz stanja i u stanje j označava se sa a_{ij}. Zbir svih verovatnoća prelaska iz jednog stanja (uključujući i sebe) je 1. Ove verovatnoće ne zavise od vremena i konstantne su. U opštem slučaju u kome je svako stanje dostižno iz bilo kog drugog stanja pretpostavlja se da je $a_{ij} > 0$ za svako i i j. Za neke druge tipove Markovljevih modela može se apriori usvojiti da je $a_{ij} = 0$ za neke specifične parove i i j.
- $E = \{e_j(k)\}$ je matrica emisionih verovatnoća ili verovatnoća emitovanja simbola k iz stanja j. U slučaju nepoštene kockarnice, ove verovatnoće su prikazane na slici 3.20. Zbir svih emisionih verovatnoća iz jednog stanja mora biti 1.

Na osnovu zadatog skrivenog Markovljevog modela $(\sum, Q, \pi, A, E)$ dobijena ispoljena sekvenca $X = X_1X_2...X_N$ može da se generiše kroz sledeći niz koraka:

1. Izabere se inicijalno stanje π_1 shodno inicijalnoj raspodeli verovatnoća π.

2. Iz stanja π_1 emituje se simbol X_1 shodno verovatnoćama pojave simbola X_1 za stanje π_1.

3. Izvrši se prelazak iz stanja π_1 u stanje π_2 na osnovu matrice verovatnoće prelaska A.

4. Iz stanja π_2 emituje se simbol X_2.

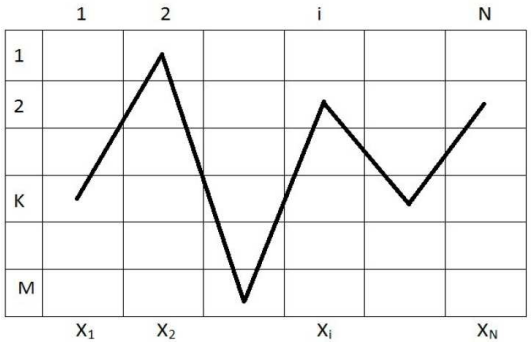
5. ...

6. Iz stanja π_{N-1} prelazi se u stanje π_N

7. Iz stanja π_N emituje se simbol X_N.

Ovaj postupak generisanja sekvenci se može grafički prikazati kao na slici 3.21, a verovatnoća generisanja sekvence se može zapisati na sledeći način:

$$P(X, \Pi) = P(\pi_1)P(X_1|\pi_1)P(\pi_2|\pi_1)P(X_2|\pi_2)...P(\pi_N|\pi_{N-1})P(X_N|\pi_N)$$



Slika 3.21: Grafički prikaz generisanja sekvence metodom skrivenih Markovljevih modela.

3.5.2 Tri osnovna problema

U želji da se formira skriveni Markovljev model koji će efikasno opisati fizički proces, kao i da se primeni već postojeći model, mogu da se pojave tri ključna problema:

- *Problem evaluacije (eng. The Evaluation Problem):* Ako je data ispoljena sekvenca $X = X_1X_2...X_N$ i ako je dat model, pitanje je kako efikasno izračunati verovatnoću da je takav model generisao takvu ispoljenu sekvencu (izračunavanje skora i evaluacija modela).

- *Problem dekodiranja (eng. The Decoding Problem):* Koja je najverovatnija "putanja" za generisanje date sekvence (dekodiranje).
- *Problem učenja (eng. The Learning Problem):* Kako mogu da se uče parametri skrivenog Markovljevog modela na datoj kolekciji sekvenci.

Rešenje problema evaluacije

Problem evaluacije predstavlja postupak kojim se računa verovatnoća generisanja neke sekvence iz poznatog modela. Ovaj se problem može tretirati i kao problem ocene u kojoj meri neka sekvenca odgovara usvojenom modelu. Ako je data putanja (sekvenca stanja) $\pi = \pi_1\pi_2...\pi_L$ i sekvenca ispoljenih odnosno emitovanih simbola $X = X_1X_2...X_L$, može se izračunati verovatnoća da se putanjom $\pi_1\pi_2...\pi_L$ generisala sekvenca $X_1X_2...X_L$ po sledećoj formuli:

$$P(X_1X_2...X_L, \pi_1\pi_2...\pi_L) = a_{0\pi_1}a_{\pi_L N}\prod_{i=1}^{L-1} a_{\pi_i\pi_{i+1}}\prod_{i=1}^{L} e_{\pi_i}(X_i)$$

Obično je poznata samo ispoljena sekvenca, ne i putanja stanja. Verovatnoća pojave ispoljene sekvence dobija se sumiranjem preko svih mogućih putanja stanja:

$$P(X_1X_2...X_L) = \sum_{\pi} P(X_1X_2...X_L, \underbrace{\pi_1\pi_2...\pi_L}_{\pi})$$

Problem koji se javlja je taj što broj putanja može da bude eksponencijalno zavisan u odnosu na dužinu sekvence. Zbog toga se koristi algoritam *Unapred (eng. Forward)* koji omogućuje efikasno izračunavanje. Ovaj algoritam se zasniva na dinamičkom programiranju. Potrebno je definisati $f_k(i)$ kao verovatnoću generisanja prvih i karaktera i dolaska u stanje k. Cilj je izračunati $f_N(L)$ kao verovatnoću generisanja cele skevnce X i dolaska u završno stanje N.

Algoritam *Unapred* se može definisati na sledeći način:

- Inicijalizacija
  - $f_0(0) = 1$, verovatnoća da smo u početnom stanju i da je generisano 0 karaktera sekvence.
  - $f_k(0) = 0$, za stanja k koja nisu tiha stanja (emituju neki karakter).
- Rekurzija

  - za emitujuća stanja ($i = 1, ..., L$):

$$f_l(i) = e_l(i) \sum_k f_k(i-1)a_{kl}$$

  - za tiha stanja:

$$f_l(i) = \sum_k f_k(i)a_{kl}$$

- Završetak:
  - Verovatnoća da smo u završnom stanju i da je generisana cela sekvenca:

$$P(X) = P(X_1...X_L) = f_N(L) = \sum_k f_k(L)a_{kN}$$

Rešenje problema dekodiranja

Problem dekodiranja predstavlja problem rešavanja skrivenosti u skrivenim Markovljevim modelima i on pokušava da da odgovor na pitanje koja logična sekvenca stanja odgovara dobijenoj ispoljenoj sekvenci. Ovaj problem se takođe može rešiti dinamičkim programiranjem korišćenjem *Viterbi* algoritma. Potrebno je definisati $v_k(i)$ kao verovatnoću najverovatnije putanje koja se završava u stanju k i pri tome generiše prvih i karaktera sekvence X. Cilj je dobiti $v_N(L)$ koje predstavlja verovatnoću najverovatnije putanje kojom se generiše cela sekvenca i koja se završava u završnom stanju. Putanja koja ima najveću vrednost verovatnoće do pozicije i je deo cele putanje koja ima najveću verovatnoću. Rešenje se može definisati rekurzivno.

Algoritam Viterbi za pronalaženje najverovatnije putanje može da se prikaže kroz sledeći niz koraka (videti sliku 3.22):

- Inicijalizacija:
  - $v_0(0) = 1$.
  - $v_k(0) = 0$, za stanja k koja nisu tiha.
- Rekurzija (pamti se najverovatnija putanja)
  - za emitujuća stanja ($i = 1, ..., L$):

$$v_l(i) = e_l(x_i) \max_k [v_k(i-1)a_{kl}]$$

$$ptr_l(i) = \operatorname{argmax}_k [v_k(i-1)a_{kl}]$$

– za tiha stanja:

$$v_l(i) = \max_k [v_k(i)a_{kl}]$$

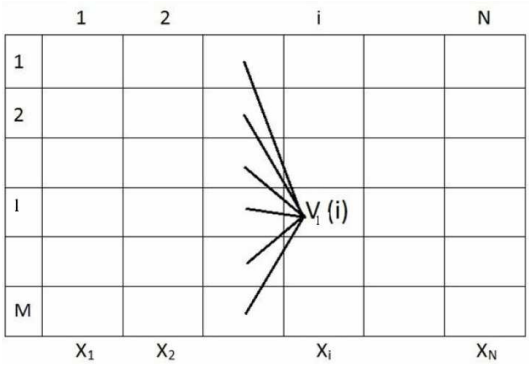
$$ptr_l(i) = \operatorname{argmax}_k [v_k(i)a_{kl}]$$

- Završavanje:

$$P(x, \pi^*) = \max_k (v_k(L)a_{kN})$$

$$\pi_L^* - \operatorname{argmax}_k (v_k(L)a_{kN})$$

Najverovatnija putanja se dobija praćenjem unazad – prate se pokazivači unazad krećući od π_L^*



Slika 3.22: Viterbi algoritam.

Rešenje problema učenja

Problem učenja se obično javlja na samom početku prilikom formiranja modela. Polazi se od skupa merenja koja su predstavljena u formi ispoljene sekvence, a cilj je da se "obuči" model tako što će se odrediti njegovi parametri tako da on najbolje odgovara prikupljenim merenjima. Neka od pitanja koja se u okviru problema 3 mogu postaviti u slučaju nepoštene kockarnice su:

- Da li i kada kockar vara?
- Kolika je verovatnoća šestice na nefer kocki?
- Kolika je verovatnoća ostalih brojeva na nefer kocki?
- Da li je fer kocka fer?

- Koliko često kockar menja kocke?

Sve ovo su nepoznati parametri čije se određivanje naziva učenje parametara. Učenje je lako ako se zna korektna putanja za svaku sekvencu u skupu za obučavanje. Nema skrivenog stanja tokom obučavanja i proces je isti kao za model Markovljevog lanca. Ako je korektna putanja za svaku sekvencu u skupu za učenje nepoznata, potrebno je razmotriti sve moguće putanje za datu sekvencu. Parametri se mogu proceniti procedurom koja prebrojava očekivani broj prelaza i emitovanja u skupu za učenje.

Algoritam za učenje parametara naziva se *Baum-Welch algoritam* a zove se i algoritam *Unapred-Unazad*. Ovaj algoritam pripada familiji algoritama maksimizovanja očekivanja, odnosno algoritama za obučavanje verovatnosnih modela u problemima koji uključuju skriveno stanje. U ovom kontekstu, skriveno stanje je putanja koja objašnjava svaku sekvencu za obučavanje.

Skica ovog algoritma se može prikazati kroz sledeći niz koraka:

- Inicijalizovati parametre modela.
- Iterirati do konvergencije.
  - Korak očekivanja: izračunati očekivani broj prelaza i emisija.
  - Korak maksimizovanja: prilagoditi parametre tako da maksimizuju verodostojnost očekivanih vrednosti.

Korak očekivanja U cilju izračunavanja očekivanog broja prelaza stanja i emitovanja karaktera, potrebno je da se izračuna verovatnoća da se simbol i generiše u stanju k ako je data sekvenca X: $P(\pi_i = k|X)$.

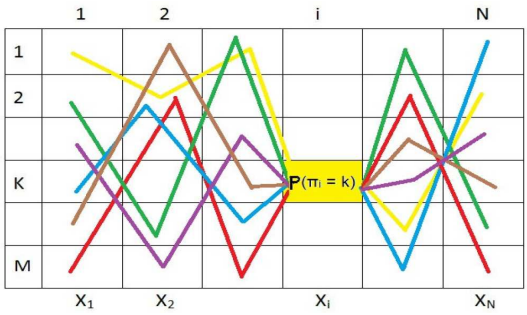
Verovatnoća generisanja X pri čemu je simbol i generisan u stanju k je

$$P(\pi_i = k, X) = P(X_1...X_i, \pi_i = k) \times P(X_{i+1}...X_L|\pi_i = k)$$

Prvi faktor je $f_k(i)$ (verovatnoća da smo u stanju k i da je generisano prvih i karaktera sekvence X) i izračunava se algoritimom *Unapred* a drugi faktor je $b_k(i)$ (verovatnoća da je generisan ostatak sekvence X kada se zna da smo u stanju k posle i karaktera) i izračunava se algoritimom *Unazad* (vidi sliku 3.23).

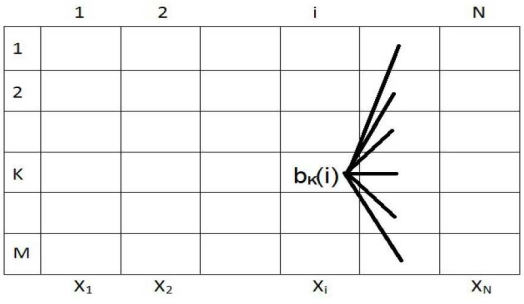
Spajajući algoritme *Unapred* i *Unazad* može da se izračuna verovatnoća generisanja sekvence X sa simbolom i generisanim u stanju k.

Algoritam *Unazad* može da se opiše kroz sledeći niz koraka (vidi sliku 3.24):



Slika 3.23: Unapred-Unazad algoritam.

- Inicijalizacija:
$$b_k(L) = a_{kN}$$
za svako stanje k koje ima prelaz u završno stanje L.
- Rekurzija $(i = L-1, ..., 0)$:
  - za emitujuća stanja:
$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$
  - za tiha stanja:
$$b_k(i) = \sum_l a_{kl} b_l(i)$$



Slika 3.24: Unazad algoritam.

Ovaj algoritam može služiti kao alternativa algoritmu Unapred za izračunavanje verovatnoće sekvence:

$$P(X) = P(X_1...X_L) = b_0(0)$$

Sada je moguće izračunati verovatnoću i-tog simbola koji se generiše u stanju k kada je data sekvenca X:

$$P(\pi_i = k|X) = \frac{P(\pi_i = k, X)}{P(X)} = \frac{f_k(i)b_k(i)}{P(X)} = \frac{f_k(i)b_k(i)}{f_N(L)}$$

Očekivani broj emitovanja karaktera c u stanju k može da se izračuna na sledeći način:

$$I_{i,j,k} = \begin{cases} 1 & \text{ako je } \pi_i = k \text{ za sekvencu } j \\ 0 & \text{u suprotnom} \end{cases}$$

$$C_{k,c} = \sum_j \sum_{\{i|x_i^j = c\}} I_{i,j,k}$$

pri čemu indeks j označava j-tu sekvencu u skupu za učenje, a i poziciju karaktera c u j-toj sekvenci x^j. Iz ovoga sledi:

$$n_{k,c} = E[C_{k,c}] = \sum_j \sum_{\{i|x_i^j = c\}} E[I_{i,j,k}]$$

$$n_{k,c} = \sum_j \frac{1}{f_N^j(L)} \sum_{\{i|x_i^j = c\}} f_k^j(i)b_k^j(i)$$

Sada može da se izračuna očekivani broj prelaza iz stanja k u stanje l:

$$n_{k\to l} = \sum_{x^j} \frac{\sum_i f_k^j(i)a_{kl}e_l(x_{i+1}^j)b_l^j(i+1)}{f_N^j(L)}$$

ili, ako je l tiho stanje:

$$n_{k\to l} = \sum_{x^j} \frac{\sum_i f_k^j(i)a_{kl}b_l^j(i)}{f_N^j(L)}$$

Korak maksimizovanja Neka je $n_{k,c}$ očekivani broj emisija karaktera c iz stanja k za skup za učenje. Potrebno je proceniti nove emisione parametre:

$$e_k(c) = \frac{n_{k,c}}{\sum_{c'} n_{k,c'}}$$

Nekal je $n_{k \to l}$ očekivani broj prelaza iz stanja k u stanje l za skup za učenje. Potrebno je proceniti novi parametar prelaza:

$$a_{kl} = \frac{n_{k \to l}}{\sum_{m} n_{k \to m}}$$

Algoritam Baum-Welch Algoritam Baum-Welch se sada može opisati kroz sledeći niz koraka:

- Inicijalizovati parametre skrivenog Markovljevog modela.
- Iterirati do konvergencije.
  - Inicijalizovati $n_{k,c}$, $n_{k \to l}$ pseudobrojačima.
  - *Korak očekivanja:* za svaku sekvencu $j = 1, ..., n$ iz skupa za učenje:
    - Izračunati $f_k(i)$ vrednosti za sekvencu j.
    - Izračunati $b_k(i)$ vrednosti za sekvencu j.
    - Dodati doprinos sekvence j brojačima $n_{k,c}$, $n_{k \to l}$.
  - *Korak maksimizovanja:* ažurirati parametre skrivenog Markovljevog modela korišćenjem $n_{k,c}$, $n_{k \to l}$.

3.5.3 Složenost izračunavanja algoritma

Za skriveni Markovljev model sa N stanja i sekvencu dužine L, vremenska složenost algoritama *Unapred*, *Unazad* i *Viterbi* je $O(N^2 L)$. Za M datih sekvenci dužine L, vremenska složenost Baum-Welch algoritma u svakoj iteraciji je $O(MN^2 L)$.

3.5.4 Postojeći sistemi i domeni primene

Sistemi zasnovani na skrivenim Markovljevim modelima imaju primenu u kriptoanalizi, prepoznavanju govora, mašinskom prevođenju, u bioinformatici (predikciji gena, poravnanju bio-sekvenci), analizi vremenskih serija, detekciji virusa i drugo. Neki od poznatijih klasifikacionih sistema ove vrste su:

- **HMMER**[17]**.** Koristi se u bioinformatici za pronalaženje homologne sekvence proteina i za poravnanja proteinske sekvence.
- **HHpred/HHsearch**. HHpred i HHsearch su slobodno dostupni sistemi za pretraživanje sekvenci proteina i predviđanje strukture proteina.
- **Jahmm**[18]**.** Ovaj sistem je implementiran u Java programskom jeziku. Jednostavan je za korišćenje i ima široku primenu. Koristi se u edukativne svrhe i u nauci jer se algoritmi mogu jednostavno menjati i prilagođavati potrebama. Podržana je implementacija Viterbi, Baum-Welch, Forward-Backward algoritma i drugo.

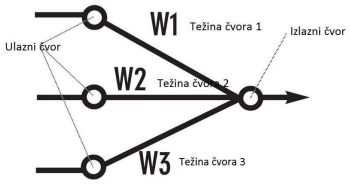
3.6 Veštačke neuronske mreže

Veštačke neuronske mreže (eng. artificial neural network, ANN) nastale su kao proizvod pokušaja da se simuliraju pravi biološki neuronski sistemi. Naime, ljudski mozak se sastoji od nervnih ćelija, takozvanih *neurona*, koji su međusobno povezani u lance, takozvane *aksone*. Kad god su neuroni stimulisani, aksoni prenose impulse iz jednog u drugi neuron. Jedan neuron je povezan sa aksonom drugih neurona kroz *dendrit* koji predstavlja mali produžetak tela neurona. Mesto spoja dendrona i aksona naziva se *sinapsa*. Neurolozi su otkrili da ljudski mozak uči tako što menja jačinu sinaptičke veze između neurona zbog ponavljajućeg stimulisanja jednim istim impulsom. Ideja je bila da se simuliraju biološki neuronski sistemi i da se dobiju veštački koji bi takođe imali mogućnost učenja. Analogno biološkim sistemima, veštačke neuronske mreže se sastoje od čvorova i veza između njih. Jedan od najjednostavnijeg oblika neuronskih mreža je *Perceptron* koji ilustruje kako modeli mogu da budu učeni da reše problem klasifikacije.

3.6.1 Perceptron

Model Perceptron prikazan je na slici 3.25. Sastoji se od dve vrste čvorova: *ulazni čvorovi*, koji predstavljaju ulazne atribute, i *izlazni čvor* (jedan po modelu) koji služi za dobijanje rezultata modela. Sami čvorovi se nazivaju *neuroni*. Svaki ulazni čvor je povezan sa izlaznim kroz vezu kojoj je dodeljena određena težina. Dakle, svaka veza ima svoju težinu, jer ona simulira jačinu sinapse u biološkim mrežama. Učenje Perceptrona se sastoji od toga da se te težine veza međusobno adaptiraju tako da oslikavaju realnu vezu između ulaznih atributa i izlazne vrednosti.

[17] http://hmmer.janelia.org/
[18] https://code.google.com/p/jahmm/



Slika 3.25: Perceptron.

Pitanje koje se postavlja jeste: *kako se dobija vrednost izlaznog čvora?* Izlazna vrednost dobija se poređenjem vrednosti nekog praga t i zbira ulaznih vrednosti u skladu sa njihovim težinama po sledećoj formuli:

$$Y = sgn(\sum_{i=1}^{k} w_i X_i - t)$$

pri čemu je Y izlazna vrednost, w_i težina i-te ulazne vrednosti, X_i je i-ta ulazna vrednost, t je prag na izlazu, k je broj ulaznih vrednosti ili atributa a *sgn* je funkcija aktivacije.

Sledeće pitanje na koje treba dati odgovor je: *kako se uči Perceptron, odnosno kako se određuju težine modela?*
Perceptron se uči iterativno, pri čemu se u svakoj iteraciji težine dodatno podešavaju sve dok se u nekoj iteraciji ne dogodi da se težine nisu promenile. Prethodno pitanje se time svodi na pitanje: *kako se težine menjaju u iteracijama?* To i jeste ključno pitanje u Perceptronu. Težine se menjaju po sledećoj formuli:

$$w^{k+1} = w^k + \gamma(y_i - y_i^k)x_i$$

pri čemu je w^k vektor težina k-te iteracije, γ je parametar poznat pod nazivom koeficijent učenja, x_i je vektor ulaznih vrednosti za i-ti primer, y_i je prava klasa tog primera, dok y_i^k je predviđena klasa i-tog primera sa težinama k-te iteracije. Ova formula zapravo štimuje težine veza u mreži sve dok Perceptron za sve podatke za učenje ne bude siguran da ih je naučio, odnosno da ih dobro klasifikuje.

Algoritam za učenje neuronskih mreža može se prikazati kroz sledeće korake:

- Inicijalizovati težine w_i.
- Podešavati vrednosti za w_i tako da izlaz iz neuronske mreže bude u

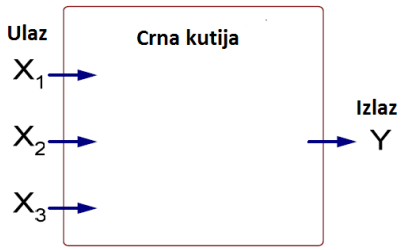
skladu sa oznakama klasa skupa za učenje. Ciljna funkcija je:

$$E = \sum_i [y_i - f(w_i, x_i)]^2$$

- Naći težine w_i' tako da se minimizuje prethodna ciljna funkcija.

Izgradnja modela Perceptron može se ilustrovati sledećim primerom prikazanim na slici 3.26:

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



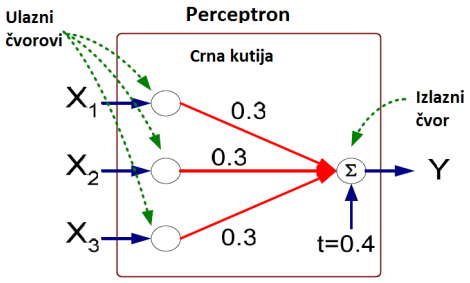
Slika 3.26: Primer izračunavanja logičke funkcije.

Primer 3.6 *Date su tri logičke promenljive x_1, x_2 i x_3. Treba dobiti izlaz y koji ima vrednost* 1 *ako bar dve ulazne promenljive imaju vrednost* 1.

Rešenje:

$$y = sgn(0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0)$$

Perceptron model rešenja prikazan je na slici 3.27.

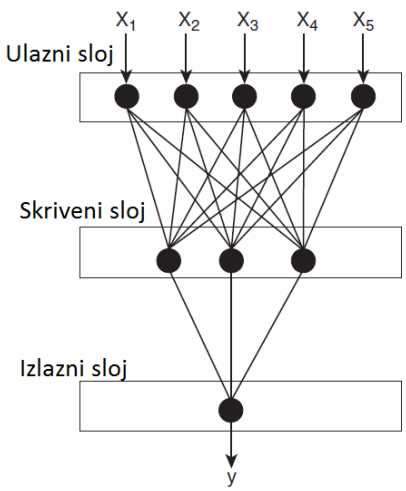


Slika 3.27: Modeliranje logičke funkcije Perceptron modelom.

Neuronske mreže imaju kompleksniju strukturu od Perceptron modela [75]:

- Mreža može biti višeslojna, što značajno komplikuje izgled mreže. Osim ulaznog i izlaznog sloja, mogu postojati i takozvani *skriveni slojevi* sa *skrivenim čvorovima*. Primer višeslojne mreže prikazan je na slici 3.28. Razlikuju se *mreže koje "idu napred"* (eng. *feed-forward*), gde su čvorovi jednog sloja povezani samo sa čvorovima sledećeg sloja i *rekurentne mreže* (eng. *recurrent*) gde čvorovi mogu biti povezani i sa čvorovima istog sloja i sa čvorovima prethodnih slojeva.

- Neuronska mreža može koristiti druge funkcije aktivacije osim *sgn* kao što je to slučaj kod Perceptrona. Primeri drugih funkcija aktivacije mogu uključiti linearne, logističke (sigmoid) funkcije, funkciju hiperbolični tangens i druge (videti sliku 3.29). Ove funkcije dozvoljavaju da skriveni i izlazni čvorovi proizvedu izlazne vrednosti koje su nelinearne u odnosu na ulazne parametre.

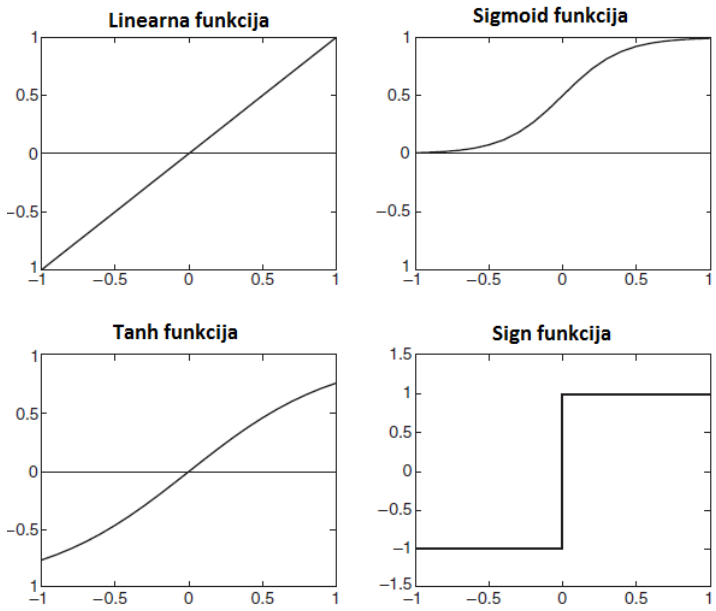


Slika 3.28: Primer višeslojne neuronske mreže.

3.6.2 Karakteristike veštačkih neuronskih mreža

Opšte karakteristike veštačkih neuronskih mreža su [75]:

- Višeslojne veštačke neuronske mreže sa bar jednim skrivenim slojem su *univerzalni aproksimatori*, odnosno mogu se koristiti za aproksimaciju bilo koje ciljne funkcije. U cilju izbegavanja prevelike prilagođenosti modela, važno je dobro odabrati odgovarajući model, odnosno broj



Slika 3.29: Tipovi funkcija aktivacije u neuronskim mrežama.

skrivenih slojeva i skrivenih čvorova kao i tip arhitekture mreže (mreža koja "ide napred" ili rekurentna mreža).

- Veštačke neuronske mreže mogu dobro da obrade suvišne atribute jer se težine automatski izračunavaju još u toku faze učenja. Iz tog razloga, težine suvišnih atributa će imati veoma male vrednosti pa samim tim i veoma mali uticaj na model.
- Neuronske mreže su veoma osetljive na prisustvo šuma u podacima za učenje.
- Učenje neuronskih mreža je vremenski zahtevan proces, pogotovu kada je broj skrivenih čvorova velik. Za razliku od toga, primeri za testiranje se mogu brzo klasifikovati.

3.6.3 Postojeći sistemi i domeni primene

Neki od postojećih klasifikacionih sistema zasnovanih na neuronskim mrežama su:

- **MATLAB Neural Net Toolbox**[19]**.** Ovaj alat predstavlja sveobuh-

[19]http://www.mathworks.com/products/neuralnet/

vatno okruženje za istraživanje, projektovanje, i simulacije u MATLAB-u, koji su zasnovani na neuronskim mrežama.

- **Decider**[20]**.** Najčešće se koristi za analizu kreditnog rizika i za otkrivanje prevara.
- **Minotaur**[21]**.** Ovo je alat od istog proizvođača kao i Decider. Ima primenu u analizi i otkrivanju prevara u telokominikacionoj industriji.
- **NeuroXL**[22]**.** Ovo je sistem za klasifikaciju jednostavnih i složenih podataka u Excel-u. Može naći poslovnu primenu, primenu u marketingu, finansijama, nauci i drugo.
- **Sharky Neural Network**[23]**.** Ovo je slobodno dostupan klasifikacioni sistem zasnovan na neruonskim mrežama. Glavna primena je u obrazovanju u cilju boljeg razumevanja klasifikacije primenom neuronskih mreža.

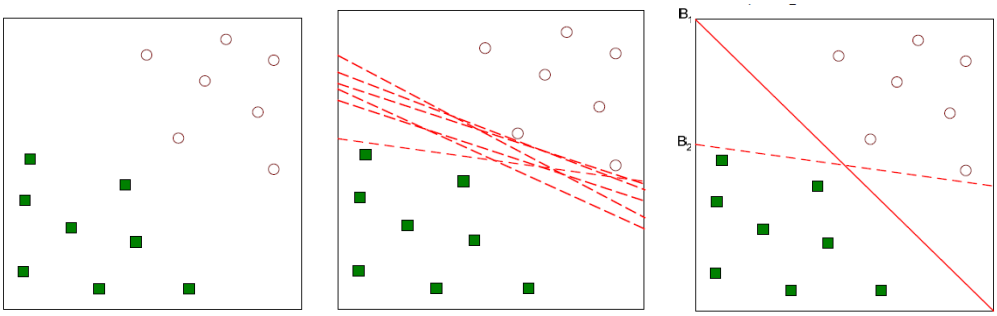
3.7 Metoda podržavajućih vektora

Metoda podržavajućih vektora (eng. support vector machine, SVM) je tehnika za klasifikaciju zasnovana na ideji vektorskih prostora. Ovu metodu su prvi predstavili Vapnik i saradnici 1992. godine [5]. Od tada ona predstavlja veoma moćan aparat koji je nadmašio druge metode u širokom rasponu primena, pre svega u primeni kod klasifikacije tekstova. Ovo je metoda za automatsko generisanje klasifikacionog modela (ciljne funkcije, klasifikatora) koji predstavlja formulu a ne skup pravila (klasa se "računa"). Osnovni algoritam definisan je za binarnu klasifikaciju.

Osnovna ideja ove metode jeste da se u vektorskom prostoru u kome su primeri (podaci) predstavljeni pronađe razdvajajuća hiper-ravan tako da su svi primeri iz iste klase sa iste strane ravni. Zadatak faze učenja metode podržavajućih vektora jeste pronaći *optimalnu* hiper-ravan koja razdvaja primere za učenje. Na slici 3.30 su predstavljeni primeri za učenje i neka od mogućih rešenja za hiper-ravan koja razdvaja te primere.

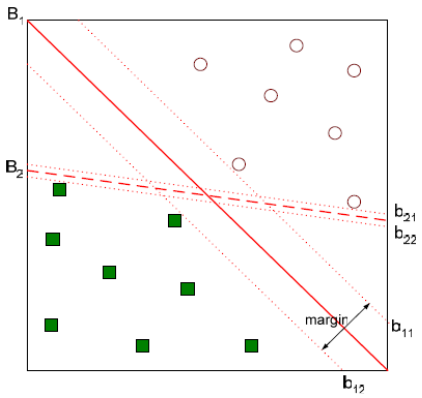
Posmatrajući ovu sliku postavlja se pitanje koje je rešenje bolje, B_1 ili B_2 i kako definisati pojam "bolje rešenje". Cilj je pronaći hiper-ravan koja *maksimizuje veličinu margine* odnosno maksimizuje rastojanje od primera

[20] www.neuralt.com
[21] www.neuralt.com
[22] http://www.neuroxl.com/
[23] http://sharktime.com/us_SharkyNeuralNetwork.html



Slika 3.30: Hiper-ravan koja razdvaja primere za učenje – neka od mogućih rešenja.

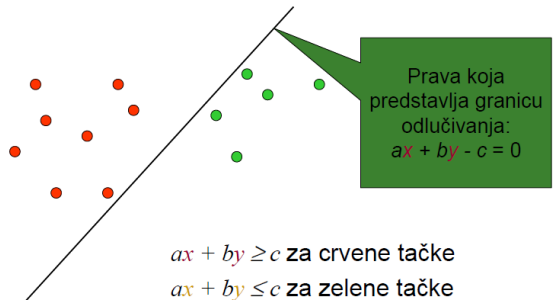
za učenje. Iz toga sledi da je B_1 bolje rešenje od B_2 (videti sliku 3.31). Ta hiper-ravan, odnosno njena jednačina, predstavlja klasifikacioni model ili klasifikator. Za primer za testiranje koga treba klasifikovati, izračunava se rastojanje od hiper-ravni i na osnovu toga se određuje klasa kojoj pripada (iznad/ispod ravni).



Slika 3.31: Hiper-ravan koja razdvaja podatke za učenje – optimalno rešenje.

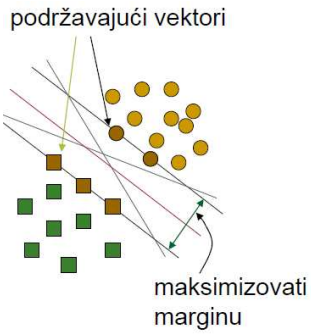
3.7.1 Linearno razdvojivi podaci

U slučaju linearno razdvojivih podataka (eng. linearly separable), hiperravan koja predstavlja granicu odlučivanja je prava. Svi primeri koji su sa jedne strane te prave pripadaju jednoj klasi a svi primeri sa druge strane prave pripadaju drugoj klasi (videti sliku 3.32).



Slika 3.32: Linearni klasifikacioni model.

Postoje jednostavni algoritmi koji mogu da odrede razdvajajuću hiper-ravan (na primer *Perceptron*), ali ne i optimalnu. Metoda podržavajućih vektora određuje optimalno rešenje koje maksimizuje razdaljinu između hiper-ravni i tačaka koje su blizu potencijalne linije razdvajanja. Intuitivno, ako nema tačaka blizu linije razdvajanja, onda će klasifikacija biti relativno laka. Cilj je pronaći ne samo hiper-ravan koja razdvaja primere već onu hiper-ravan sa maksimalnom marginom. Rezultat toga je da je razdvajajuća hiper-ravan potpuno određena specifičnim podskupom primera za učenje, koji se zovu podržavajući (potporni) vektori, po čemu je metoda i dobila ime (videti sliku 3.33).



Slika 3.33: Linearni klasifikacioni model – podržavajući vektori.

Formalizacija problema

Neka je dat prostor dimenzije n. Svaki primer i, $i = 1, ..., N$ neka je predstavljen vektorom $\mathbf{x}_i = (x_1, ...x_n)$ sa pridruženim oznakama klase $y_i \in \{1, -1\}$

(binarna klasifikacija). Jednačina hiper-ravni može se definisati izrazom:

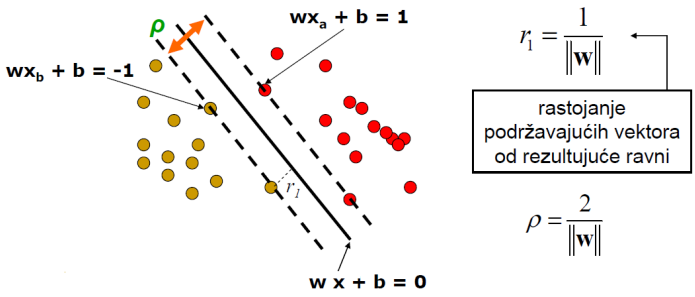
$$\mathbf{w}^T\mathbf{x} + b = 0$$

i ona je potpuno određena parametrima $\mathbf{w}$ i b. $\mathbf{x}$ je primer za učenje (ima ih ukupno N). Parametar $\mathbf{w}$ predstavlja vektor težina i određuje smer hiper-ravni, dok je parametar b pomeraj, i određuje udaljenost hiper-ravni od centra koordinatnog sistema. Potrebno je odrediti kanonske vrednosti za $\mathbf{w}$ i b.

Jednačina rastojanja tačke $\mathbf{x}$ od ravni Π data je izrazom:

$$d(\mathbf{x}, \Pi) = r = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$$

Važno je naglasiti da vrednost ovog rastojanja može biti i negativna. Margina ρ je širina razdvajanja između klasa koju treba maksimizovati (videti sliku 3.34). Kanonske vrednosti za $\mathbf{w}$ i b određuju se tako da je razdaljina najbližih



Slika 3.34: Linearni klasifikacioni model – margina.

tačaka (podržavajućih vektora) jednaka 1 po apsolutnoj vrednosti. Rastojanje podržavajućih vektora od rezultujuće ravni biće $r_1 = \frac{1}{\|\mathbf{w}\|}$ a debljina margine biće $\rho = 2r_1 = \frac{2}{\|\mathbf{w}\|}$ što predstavlja prvi uslov koji se postavlja.

Za svaku tačku $(\mathbf{x}_i, y_i)$, uslov razdvajanja može da se formuliše na sledeći način:

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1 \text{ ako } y_i = 1$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \text{ ako } y_i = -1$$

Dakle, problem se može formalizovati na sledeći način:

Naći $\mathbf{w}$ i b tako da se maksimizuje ρ uz uslov $\mathbf{w}^T\mathbf{x}_i + b \geq 1$ ako je $y_i = 1$, odnosno $\mathbf{w}^T\mathbf{x}_i + b \leq -1$ ako je $y_i = -1$. Kako važi $min\|\mathbf{w}\| = max\frac{1}{\|\mathbf{w}\|}$, problem se može formalizovati i kao: Naći $\mathbf{w}$ i b tako da se minimizuje $f(w) = \frac{1}{2}\|\mathbf{w}\|^T\|\mathbf{w}\|$ (uslov maksimalne margine) uz uslov $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$ (uslov razdvajanja).

Matematičko rešenje problema:

Ovaj problem predstavlja poznati kvadratni optimizacioni problem uz linearne uslove za koga postoji više metoda za rešavanje. Jedno od rešenja je pomoću Lagranžeovih multiplikatora [9], pri čemu se formuliše takozvani dualni problem.

Oblik rešenja je

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

pri čemu su $(\mathbf{x}_i, y_i)$ primeri za treniranje. Potrebno je naći $\alpha_1, ..., \alpha_N$ takve da je $Q(\alpha) = \sum \alpha_i - \frac{1}{2}\sum\sum \alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j$ maksimalno uz uslove $\sum \alpha_j y_i = 0$ i $\alpha_i \geq 0$ za sve α_i.

Rešenje je oblika

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$
$$b = y_k - \mathbf{w}\mathbf{x}_k$$

za svako $\mathbf{x}_k$ za koje je $\alpha_k \neq 0$. Podržavajući vektori su svi za koje je $\alpha_k \neq 0$ dok ostali ne učestvuju u rešenju.

Dakle, model klasifikacije ima sledeći oblik:

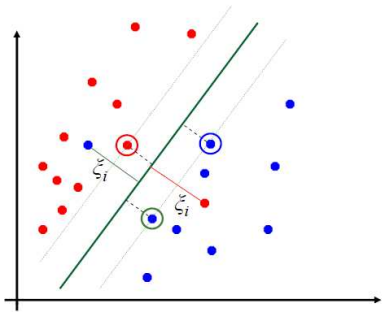
$$f(x) = sign(\sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b)$$

Može se zaključiti da je model klasifikacije predstavljen skupom koeficijenata α_i i skupom podržavajućih vektora $\mathbf{x}_i$. Primećuje se da je za klasifikaciju svakog elementa x potrebno izračunati skalarni proizvod sa podržavajućim vektorima.

Metoda podržavajućih vektora može da se uopšti i na klasu problema kada podaci nisu linearno razdvojivi i to na dva načina [57]: klasifikacija sa mekom marginom i klasifikacija pomoću kernela.

3.7.2 Klasifikacija sa mekom marginom

Ako skup za učenje nije linearno razdvojiv potrebno je uvesti promenljive ξ_i koje će tolerisati male greške u fazi učenja a kasnije i u fazi testiranja (videti sliku 3.35).



Slika 3.35: Klasifikacija sa mekom marginom.

Nova formulacija problema glasi: Naći w i b tako da se minimizuje

$$L(\mathbf{w}) = \frac{\|\vec{\mathbf{w}}\|^2}{2} + C\sum_{i=1}^{N}\xi_i^k$$

uz uslov

$$f(\vec{\mathbf{x}_i}) = \begin{cases} 1, & \text{ako je } \vec{\mathbf{w}} \circ \vec{\mathbf{x}_i} + b \geq 1 - \xi_i, \\ -1, & \text{ako je } \vec{\mathbf{w}} \circ \vec{\mathbf{x}_i} + b \leq -1 + \xi_i. \end{cases}$$

pri čemu je C parametar koji kontroliše proces odnosno balansira između margine i greške prilikom učenja (minimizovaće grešku na račun ne baš maksimalne margine). Za model klasifikacije dobija se:

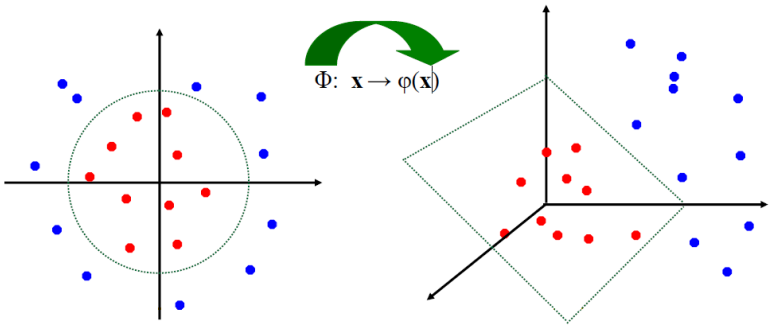
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

pri čemu treba odabrati prag odlučivanja p takav da važi:
$f(\mathbf{x}) > p$ ako element pripada klasi,
$f(\mathbf{x}) < p$ ako element ne pripada klasi, dok je
$f(\mathbf{x}) = p$ nedefinisan slučaj.

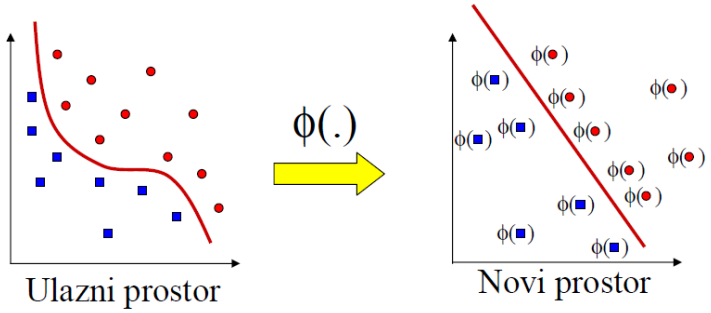
3.7.3 Kernel funkcije (funkcije jezgra)

U slučaju linearno nerazdvojivih podataka, drugi način rešenja problema je da se osnovni vektorski prostor u kome je skup za učenje linearno nerazdvojiv preslika u neki višedimenzionalni prostor u kome je skup za učenje linearno razdvojiv, pomoću preslikavanja $\Phi : \mathbf{x} \longrightarrow \varphi(\mathbf{x})$ (videti slike 3.36 i 3.37).

Umesto skalarnog proizvoda (mera sličnosti dva vektora) uvodi se kernel funkcija koja odgovara skalarnom proizvodu u preslikanom prostoru (prostru



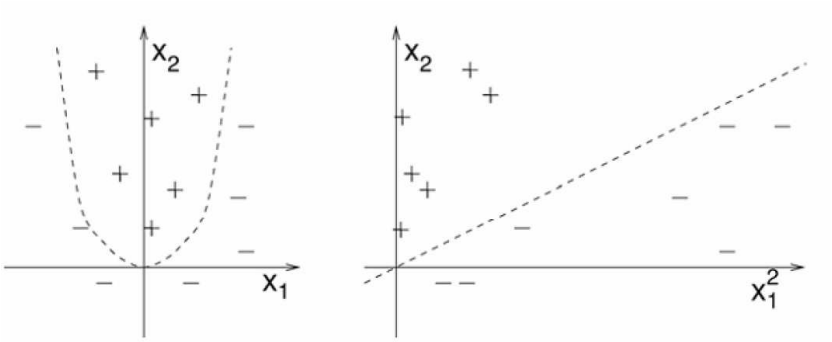
Slika 3.36: Nelinearni SVM – preslikavanje iz jednog u drugi prostor.



Slika 3.37: Nelinearni SVM – preslikavanje iz jednog u drugi prostor.

veće dimenzije) [64]. Linearni klasifikatori su zasnovani na ideji skalarnog proizvoda $K(x_i, x_j) = x_i^T x_j$ u osnovnom (ulaznom) prostoru. Ako se svaka tačka preslika u prostor veće dimenzije koristeći transformaciju $\Phi : x \longrightarrow \varphi(x)$, skalarni proizvod postaje: $K(x_i, x_j) = \varphi(x_i)\varphi(x_j)$.

Jedan primer preslikavanja Φ dat je slikom 3.38. Ulazni (originalni) prostor je 2-dimenzionalan: $x = (x_1, x_2)$ a preslikani prostor je 6-dimenzionalan: $\varphi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$. Postavlja se pitanje: koji kernel odgovora ovom preslikavanju Φ i dali važi $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$? Može se pokazati da je $K(x_i, x_j) = (1 + x_ix_j)^2$ tražena funkcija:



Slika 3.38: Nelinearni SVM – primer preslikavanja.

$$\begin{aligned} K(x_i, x_j) &= (1 + x_i x_j)^2 \\ &= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} \\ &= [1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}] \circ \\ &\quad [1, x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}] \\ &= \varphi(x_i) \circ \varphi(x_j) \end{aligned}$$

gde je $\varphi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$

Dakle, ako se preslika originalni 2-dimenzionalni prostor u 6-dimenzionalni koristeći funkciju $\varphi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$ u tom prostoru dobija se skalarni proizvod $\varphi(x_i) \circ \varphi(x_j)$ koji se računa preko originalnih koordinata koristeći kernel $(1 + x_i x_j)^2$.

Novodobijeni prostor ne mora biti poznat eksplicitno (ne mora da bude predstavljen jednačinom) već je dovoljno da bude poznato kako se računa njegov skalarni proizvod. Jedino je bitno da preslikavanje K odgovara skalarnom proizvodu u nekom (novom) prostoru.

Dakle, kernel funkcije se koriste da bi se dati ne-linerno razdvojiv problem transformisao u linearno razdvojiv. U prostoru veće dimenzije, mnogo je verovatnije da će se dobiti linearno razdvajanje. Dakle, konstruiše se hiperravan u prostoru veće dimenzije, a kada se ta ravan preslika nazad, dobija se nelinearno razdvajanje u početnom prostoru.

Problem koji se nameće je kako odabrati preslikavanje K odnosno kernel koji odgovara skalarnom proizvodu u nekom novom prostoru.

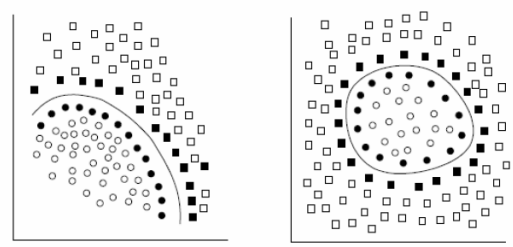
Neki primeri uobičajenih kernela su (videti sliku 3.39):

$$\text{Polinomialan: } K(x_i, x_j) = (x_i x_j + c)^q$$

$$\text{RBF(radial basis function): } K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

$$\text{Simnoide: } K(x_i, x_j) = tanh(\alpha x_i x_j - b)$$

Postoji teorija o tome da li postoji i kako kontruisati ispravan kernel za dati problem. Obično se koriste uobičajeni kerneli ili njihova kombinacija. Ono što treba naglasiti jeste da su kerneli zatvoreni za linearne kombinacije, množenje skalarom, međusobno množenje i drugo.



Slika 3.39: Razdvajajuća ravan: polinomijalni i RBF kernel.

Postoji matematička teorija (teorema Mercera) koja definiše uslove koje data funkcija treba da zadovolji da bi predstavljala sklarni proizvod u nekom vektorskom prostoru: simetričnost, pozitivna definitnost i tako dalje. Svaka funkcija koja zadovolji te uslove može da bude korišćena kao kernel. Navedeni primeri kernel funkcija su dovoljni u većini primena (naročito ako se uzme u obzir zatvorenost).

3.7.4 Karakteristike metode podržavajućih vektora

Metoda podržavajućih vektora važi za jednu od najuspešnijih (ili bar najkorišćenijih) klasifikacionih tehnika. Posebno je upotrebljiva u situacijama kada je broj "dimenzija" podataka velik. Nalazi primenu u klasifikaciji teksta, prepoznavanju rukopisa, klasifikaciji slika i tako dalje. Prilagođena je za baratanje velikim količinama podataka. Prilično je otporna na šum u podacima koji je neizbežna posledica bilo kog eksperimentalnog merenja. Međutim, tačnost metode je ograničena i prilično zavisi od izobora nekih parametara. Takođe, metoda je predviđenja za klasifikaciju samo na dve klase, mada su razvijeni različiti pristupi koji omogućavaju klasifikaciju na više od dve klase.

3.7.5 Postojeći sistemi i domeni primene

Neki od najpoznatijih sistema za klasifikaciju zasnovanih na metodi podržavajućih vektora su:

- **LIBSVM**[24] [8]. Ovo je popularan slobodno dostupan alat napisan u C++ programskom jeziku. Integrisan je u druge slobodno dostupne alate uključujući GATE (eng. General Architecture for Text Engineering)[25] i KNIME (eng. Konstanz Information Miner)[26].

- **SVM-light**[27]**.** Ovo je popularna implementacija Vapnikove[82] metode podržavajućih vektora autora Thorsten Joachims-a. Implementiran je u C-u. Koristi brze algoritme optimizacije opisane u [31], radi sa nekoliko hiljada podržavajućih vektora i sa nekoliko stotina hiljada primera za učenje. Podržava standardne kernel funkcije i daje mogućnost definisanja novih kernela.

- **LS-SVMlab**[28] (eng. Least squares SVM). Algoritam na kome je zasnovan ovaj alat predstavili su Suykens and Vandewalle u radu [73].

Klasifikacioni sistemi zasnovani na podržavajućim vektorima imaju značajnu primenu u klasifikaciji teksta i slika, u bioinformatici (klasifikaciji proteina, klasifikaciji kancera), prepoznavanju rukom napisanih karaktera i drugo.

[24] http://www.csie.ntu.edu.tw/∼cjlin/libsvm
[25] http://gate.ac.uk/
[26] http://www.knime.org/
[27] http://svmlight.joachims.org/
[28] http://www.esat.kuleuven.be/sista/lssvmlab/

4. Nove metode klasifikacije

4.1 Metoda zasnovana na n-gramima

Metoda zasnovana na n-gramima koja će biti predstavljena u ovom radu, u osnovi se zasniva na radu Kešelja i njegovih kolega [38] koji su se bavili problemom određivanja autorstva teksta. Predstavljanje klasa i dokumenata u okviru ove metode zasnovano je na profilima. Svaki autor (klasa) predstavljen je profilom koji čini uređeni skup parova (x_1, f_1), (x_2, f_2),..., (x_L, f_L) L najfrekventnijih n-grama x_i i njihovih normalizovanih frekvencija f_i. Autorstvo teksta se određuje na osnovu sličnosti tog teksta, odnosno njegovog profila, i profila svih predstavljenih autora (klasa). Sličnost između dva profila određuje se pomoću mere različitosti. Tekst se dodeljuje onom autoru (klasi) sa kojim ima najmanju vrednost mere različitosti, odnosno sa kojim je najsličniji.

4.1.1 n-Grami

Definicija 4.1 *Ako je data niska simbola $S = s_1 s_2 ... s_N$ nad azbukom $\mathcal{A}$, gde su N i n pozitivni celi brojevi ($n \leq N$), n-gram niske S je bilo koja podniska susednih simbola dužine n. Niska $(s_i, s_{i+1}, ..., s_{i+n-1})$ predstavlja i-ti n-gram niske S.*

Nad azbukom $\mathcal{A}$, može se definisati ukupno $|\mathcal{A}|^n$ različitih n-grama, pri čemu je $|\mathcal{A}|$ veličina (kardinalnost) azbuke $\mathcal{A}$.

Ovako predstavljeni n-grami mogu biti definisani na nivou reči, karaktera ili bajta. Neka je $\mathcal{A}$ latinična azbuka a $S =$ "*Mislim, dakle postojim*!" niska nad tom azbukom $\mathcal{A}$. Primeri n-grama nad ovako definisanom niskom su:

- n-grami na nivou reči:
  - 1-grami: Mislim; dakle; postojim
  - 2-grami: Mislim dakle; dakle postojim
  - 3-grami: Mislim dakle postojim

- n-grami na nivou karaktera:
  - 1-grami: M; I; S; L; I; M; ,; _; D; A; K; L; E; _; P; O; S; T; O; J; I; M
  - 2-grami: MI; IS; SL; LI; IM; M_; _D; DA; AK; KL; LE; E_; _P; PO; OS; ST; TO; OJ; JI; IM; M_
  - 3-grami: MIS; ISL; SLI; LIM; IM_; M_D; _DA; DAK; AKL; KLE; LE_; E_P; _PO; POS; OST; STO; TOJ; OJI; JIM; IM_
- n-grami na nivou bajta:
  - 1-grami: M; i; s; l; i; m; ,; _; d; a; k; l; e; _; p; o; s; t; o; j; i; m; !;
  - 2-grami: Mi; is; sl; li; im; m,; ,_; _d; da; ak; kl; le; e_; _p; po; os; st; to; oj; ji; im; m!
  - 3-grami: Mis; isl; sli; lim; im,; m,_; ,_d; _da; dak; akl; kle; le_; e_p; _po; pos; ost; sto; toj; oji; jim; im!

Iako predstavljanje teksta na nivou reči deluje kao logično rešenje, zanemaruje se činjenica da kod nekih jezika kao što je kineski, ne postoji eksplicitna oznaka za razdvajanje reči pa je proces razdvajanja reči sam po sebi prilično komplikovan. U slučaju jezika nad latiničnom azbukom, n-grami na nivou bajta i nivou karaktera su veoma slični s obzirom na činjenicu da je jedan karakter obično predstavljen jednim bajtom. Jedina razlika je u tome što se kod n-grama na nivou karaktera najčešće ignorišu cifre, znakovi interpunkcije, ne-štampajući karakteri i ne pravi se razlika između velikih i malih slova, dok kod n-grama na nivou bajta u obzir se uzimaju svi, štampajući i neštampajući karakteri, i tekst se tretira prosto kao niz bajtova. U slučaju azijskih jezika, jedan karakter je obično predstavljen sa dva bajta (zavisno od kodne šeme koja se koristi), pa 75% n-grama na nivou bajta uključuje polovine karaktera (svi n-grami neparne dužine i polovina n-grama parne dužine).

Prednosti i nedostaci n-grama u obradi prirodnog jezika

Kada se koriste u procesu obrade prirodnih jezika, neke od dobrih osobina koje n-grami pokazuju su [78]:

- Robusnost: relativna neosetljivost na pravopisne greške.
- Kompletnost: azbuka znakova je unapred poznata.
- Nezavisnost od domena: nezavisnost od jezika i sadržaja.

- Efikasnost: izvršavanje u jednom prolazu i
- Jednostavnost: ne zahteva se nikakvo lingvističko predznanje.

Osnovni problem kod korišćenja n-grama je eksponencijalni broj njihovih mogućnosti u odnosu na kardinalnost azbuke. Ako je $\mathcal{A}$ azbuka engleskog jezika i ako se u razmatranje uključi i znak za prazninu, onda je $|\mathcal{A}| = 27$. Ako se pravi razlika između malih i velikih slova i ako se u razmatranje uključe cifre, onda je $|\mathcal{A}| = 63$. Jasno je da će mnogi algoritmi sa n-gramima biti veoma skupi sa stanovišta izračunljivosti već za $n = 5$ ili $n = 6$ (na primer, $63^5 \approx 10^9$).

Primena n-grama

Korišćenje modela i tehnika zasnovanih na n-gramima u procesu obrade prirodnih jezika pokazalo se kao efikasan pristup. Ovaj pristup našao je primenu u okviru zadatka pretraživanja informacija [12], kompresije teksta [92], otkrivanja i ispravljanja pravopisnih grešaka [96], identifikacije jezika na kome je tekst napisan [77], otkrivanje autorstva teksta [38] i drugo. Ovaj pristup se pokazao efikasan i u oblastima koje nisu povezane sa obradom prirodnih jezika kao na primer predstavljanje muzike [16], klasifikacija proteina [69] i drugo.

Cilj ovog rada između ostalog je ispitivanje efikasnosti ove metode primenjene na problem tematske klasifikacije teksta.

4.1.2 Procedura klasifikacije

Postupak klasifikacije metodom zasnovanom na n-gramima realizuje se kroz sledeći niz koraka:

1. Sakupiti kolekciju klasifikovanih tekstualnih dokumenata u korpus i podeliti ih na skup za učenje i skup za testiranje.
2. U okviru skupa za učenje, nadovezati jedan na drugi sve dokumente koji pripadaju istoj klasi, tako da svaka klasa bude predstavljenja samo jednim dokumentom.
3. Za svaki dokument klase i svaki dokument za testiranje, konstruisati njihove profile:
   (a) Izabrati određenu vrednost za parametar n (na primer 6-grami, 7-grami i tako dalje).

(b) U okviru svakog dokumenta izdvojiti n-grame za prethodno izabranu vrednost od n.

(c) Izračunati normalizovane frekvencije n-grama (broj pojavljivanja određenog n-grama podeljen ukupnim brojem pojavljivanja svih n-grama iste dužine u tom dokumentu), za svaki n-gram.

(d) Izlistati sve takve n-grame u opadajućem redosledu prema frekvenciji. Dakle, prvo treba da budu izlistani najfrekventniji n-grami. Broj mogućih n-grama će biti različit u zavisnosti od dužine tekstualnog dokumenta i vrednosti parametra n.

(e) Izabrati dužinu profila L, odnosno broj najfrekventnijih n-grama koji će biti razmatrani i koji će određivati profil.

4. Za svaki dokument za testiranje, uporediti njegov profil sa profilima svih klasa:

(a) Izračunati meru različitosti između profila dokumenta za testiranje i profila svih klasa pojedinačno.

(b) Pridružiti dokument za testiranje onoj klasi sa kojom je najsličniji odnosno ima najmanju vrednost mere različitosti.

U okviru ove procedure veoma značajnu ulogu ima mera različitosti koja se koristi u cilju određivanja sličnosti dokumenta sa klasom, i donošenja odluke da li neki dokument pripada određenoj klasi ili ne.

4.1.3 Mere različitosti

Mera različitosti d je funkcija koja preslikava Dekartov proizvod skupa profila Π u skup pozitivnih realnih brojeva. Simbolički, $d : \Pi \times \Pi \longrightarrow R$. Ova funkcija treba da reflektuje sličnost između dva profila i treba da ispunjava sledeće uslove:

- $d(\mathcal{P}, \mathcal{P}) = 0$
- $d(\mathcal{P}_1, \mathcal{P}_2) = d(\mathcal{P}_2, \mathcal{P}_1)$
- Vrednost $d(\mathcal{P}_1, \mathcal{P}_2)$ treba da bude mala ako su $\mathcal{P}_1$ i $\mathcal{P}_2$ slični.
- Vrednost $d(\mathcal{P}_1, \mathcal{P}_2)$ treba da bude velika ako su $\mathcal{P}_1$ and $\mathcal{P}_2$ različiti.

pri čemu su $P, P_1 i P_2$ proizvoljni profili iz skupa Π. Poslednja dva uslova su informativnog karaktera i nisu striktno definisani.

U ovom radu testirane su mnogobrojne mere različitosti. Mera predstavljena u radu Kešelja [38] ima formu relativnog rastojanja i može se predstaviti formulom:

$$d_1(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{2 \cdot (f_1(n) - f_2(n))}{f_1(n) + f_2(n)} \Big)^2 \tag{4.1}$$

gde su f_1 i f_2 frekvencije određenog n-grama n u profilu klase $\mathcal{P}_1$ i profilu dokumenta $\mathcal{P}_2$, redom.

U radu Tomovića [78], osim mere Kešelja predstavljenje su i neke nove mere različitosti. Neke od njih predstavljaju samo varijaciju mere Kešelja. Umesto da se razlika frekvencija deli aritmetičkom sredinom, može se deliti geometrijskom, harmonijskom ili kvadratnom sredinom. Takođe, neki elementi u sumi mogu biti kvadrirani a mogu se računati i apsolutne vrednosti razlike frekvencija. Primeri takvih mera su:

$$d_2(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{2|f_1(n) - f_2(n)|}{f_1(n) + f_2(n)} \tag{4.2}$$

$$d_3(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)} + 1} \Big)^2 \tag{4.3}$$

$$d_4(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) f_2(n)} + 1} \tag{4.4}$$

U merama d_3 i d_4 dodaje se aditivna konstanta 1 u imeniocu kako bi se izbegao slučaj da je imenilac jednak nuli kada je neka od frekvencija jednaka nuli.

Sledeće dve mere zasnovane su na harmonijskoj sredini:

$$d_5(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \Big(\frac{(f_1(n) - f_2(n))(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)} \Big)^2 \tag{4.5}$$

$$d_6(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \frac{|f_1(n) - f_2(n)|(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)} \tag{4.6}$$

Mere zasnovane na geometrijskoj sredini bez aditivne konstante prikazane su sledećim formulama:

$$d_7(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \Big(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)}} \Big)^2 \tag{4.7}$$

$$d_8(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) f_2(n)}} \tag{4.8}$$

Sledeće dve mere su konstruisane kao linearne kombinacije linearne i kvadratne razlike:

$$d_9(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} (A|f_1(n) - f_2(n)| + B|f_1(n)^2 - f_2(n)^2|) \tag{4.9}$$

pri čemu su A i B konstante sa vrednostima $A = 100$ i $B = 1$.

$$d_{10}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} (A|f_1(n) - f_2(n)| + B|f_1(n)^2 - f_2(n)^2|) \tag{4.10}$$

pri čemu su A i B konstante sa vrednostima $A = 1000$ i $B = 0.1$.

Mere zasnovane na kvadratnoj sredini su:

$$d_{11}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{\sqrt{2}(f_1(n) - f_2(n))}{\sqrt{f_1(n)^2 + f_2(n)^2}} \Big)^2 \tag{4.11}$$

$$d_{12}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{\sqrt{2}|f_1(n) - f_2(n)|}{\sqrt{f_1(n)^2 + f_2(n)^2}} \tag{4.12}$$

Sledeća mera ukazuje na uticaj aditivne konstante na geometrijsku sredinu:

$$d_{13}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)} + 10} \Big)^2 \tag{4.13}$$

Na kraju, u radu Tomovića [78] razmatrana je i mera zasnovana na Euklidovom rastojanju,

$$d_{14}(\mathcal{P}_1, \mathcal{P}_2) = \sqrt{\sum_{n \in profile} (f_1(n) - f_2(n))^2} \tag{4.14}$$

kao i mere zasnovane na Manhattan rastojanju:

$$d_{15}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} |f_1(n) - f_2(n)| \tag{4.15}$$

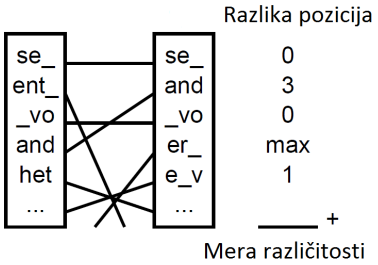
$$d_{16}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{2\sum_{n\in profile} f_1(n)f_2(n)}{\sum_{n\in profile} f_1(n)^2 + \sum_{n\in profile} f_2(n)^2} \tag{4.16}$$

$$d_{17}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n\in profile} f_1(n)f_2(n)}{\sum_{n\in profile} f_1(n)^2 + \sum_{n\in profile} f_2(n)^2 - \sum_{n\in profile} f_1(n)f_2(n)} \tag{4.17}$$

$$d_{18}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n\in profile} f_1(n)f_2(n)}{\sqrt{(\sum_{n\in profile} f_1(n)^2)(\sum_{n\in profile} f_2(n)^2)}} \tag{4.18}$$

$$d_{19}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n\in profile} f_1(n)f_2(n)}{min((\sum_{n\in profile} f_1(n)^2), (\sum_{n\in profile} f_2(n)^2))} \tag{4.19}$$

Osim mera predstavljenih u radu Tomovića i drugih [78], razmatrana je i mera koju je predstavio Cavnar u radu [7]. Ovo je jednostavna mera zasnovana na razlici pozicija n-grama u profilu dokumenta za testiranje i profilu klase, nazvana dRP. Način izračunavanja ove mere prikazan je na slici 4.1. Za svaki n-gram u profilu dokumenta za testiranje, pronalazi se isti takav n-gram u profilu klase i onda se izračunava razlika njihovih pozicija u profilima. Ako profil klase ne sadrži isti takav n-gram, onda se kao razlika njihovih pozicija uzima neka maksimalna vrednost, obično dužina profila L, odnosno ukupan broj n-grama u profilu. Sumiranjem vrednosti razlika pozicija za sve n-grame u profilu dokumenta, dobija se vrednost mere različitosti dRP između tog dokumenta i odabrane klase.



Slika 4.1: Izračunavanje mere različitosti dRP

U ovom radu predstavljena je i mera zasnovana na simetričnoj razlici. Ona predstavlja broj n-grama koji se pojavljuju u jednom ali ne i u drugom profilu.

$$dSR(\mathcal{P}_1, \mathcal{P}_2) = |\mathcal{P}_1 \triangle \mathcal{P}_2| \tag{4.20}$$

Ova mera kao i mera koju je predstavio Cavnar ne uključuje u svoje izračunavanje normalizovane frekvencije n-grama. Time se dobijaju pojednostavljeni profili koji se sastoje od L najfrekventnijih n-grama i ne uključuju nikakve informacije o frekvencijama n-grama. Frekvencije se koriste samo u procesu sortiranja n-grama a samim tim i određivanju L najfrekventnijih n-grama.

4.1.4 Modifikacije metode zasnovane na n-gramima

Jedno moguće unapređenje n-gramske metode, koje do sada nije razmatrano, predstavlja uvođenje težinskih faktora prilikom konstruisanja profila klase. Na način koji će biti opisan kasnije, n-gramima iz profila klase dodeljuju se težine prema značaju koji imaju za tu klasu. Imajući ovo u vidu, opisana metoda klasifikacije može biti modifikovana na dva načina:

1. *Modifikacija na nivou mere različitosti*, koja se zasniva na uključivanju težinskih faktora u proces izračunavanja mere različitosti između dva profila. Svaka mera koja u svom izračunavanju uključuje normalizovanu frekvenciju n-grama, može da se modifikuje tako što se normalizovana frekvencija svakog n-grama iz profila klase množi težinskim faktorom koji je dodeljen tom n-gramu.

2. *Modifikacija na nivou profila klase*, koja se zasniva na izbacivanju svih n-grama iz profila klase kojima je dodeljena težina manja od neke unapred zadate vrednosti. U okviru ove modifikacije koriste se originalne mere različitosti iz osnovne varijante metode.

U okviru svake od ovih modifikacija, profil dokumenta za testiranje se konstruiše na isti način kao i u osnovnoj varijanti metode. Osnovna razlika se, između ostalog, oglada u postupku konstruisanja profila klase. Nakon što se izabere vrednost za parametar n i nakon što se u opadajućem redosledu prema frekvenciji pojavljivanja izlistaju svi n-grami dužine n sa njihovim normalizovanim frekvencijama, u postupku konstruisanja profila klase primenjuju se sledeći koraci:

- Izabrati određenu vrednost za parametar $LTezine$ koji će da predstavlja broj najfrekventnijih n-grama na osnovu kojih se izračunava težina n-grama.

- Za svaki n-gram koji pripada grupi od $LTezine$ najfrekventnijih n-grama, vrši se izračunavanje njegovog težinskog faktora na osnovu značaja koji ima za tu klasu. Ovi težinski faktori mogu da se računaju na više načina a neki od njih su:

$$tezina(x) = |C| - c_f + 1 \tag{4.21}$$

$$tezina(x) = \frac{|C|^2}{c_f^2} \tag{4.22}$$

$$tezina(x) = log(\frac{|C|}{c_f} + 1) \tag{4.23}$$

pri čemu je $|C|$ ukupan broj klasa koji se razmatra u procesu klasifikacije a c_f je broj onih klasa kod kojih se n-gram x pojavljuje među prvih $LTezine$ najfrekventnijih n-grama. Na primer, u slučaju klasifikacije na tri klase C_1, C_2 i C_3 i izračunavanja težine po formuli 4.21, n-gramu x iz klase C_1 će biti dodeljene vrednosti po sledećem principu:

$$tezina1(x) = \begin{cases} 1, & \text{ako se } x \text{ pojavljuje u sve tri klase,} \\ 2, & \text{ako se } x \text{ pojavljuje u još jednoj od klasa } C_2 \text{ i } C_3, \\ 3, & \text{ako se } x \text{ pojavljuje samo u klasi } C_1. \end{cases}$$

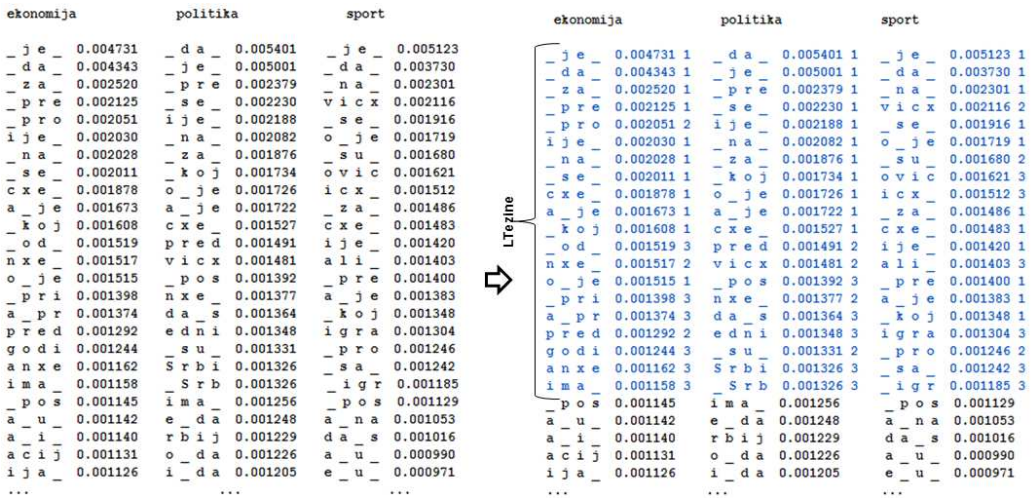
Ilustracija ovog koraka prikazana je na slici 4.2 za klase Ekonomija, Politika i Sport korpusa Ebart-3, za vrednosti parametara $n = 4$ i $LTezine = 20$. Prva kolona u okviru opisa svake klase predstavlja listu n-grama koji pripadaju toj klasi, druga kolona predstavlja njihove normalizovane frekvencije a treća predstavlja težinske faktore dodeljene tim n-gramima.

Nakon ovako definisanih težinskih faktora, profil klase se dalje može definisati na dva načina, u zavisnosti od toga koja se od modifikacija metode primenjuje.

U slučaju *modifikacije na nivou mere različitosti*, profil klase se dobija primenom sledećeg koraka:

- Izabrati dužinu profila L, odnosno broj najfrekventnijih n-grama koji će biti razmatrani i koji će određivati profil klase, pri čemu je $L \leq LTezine$. Ovaj postupak je ilustrovan na slici 4.3.

U slučaju *modifikacije na nivou profila klase*, profil klase se dobija primenom sledećih koraka:

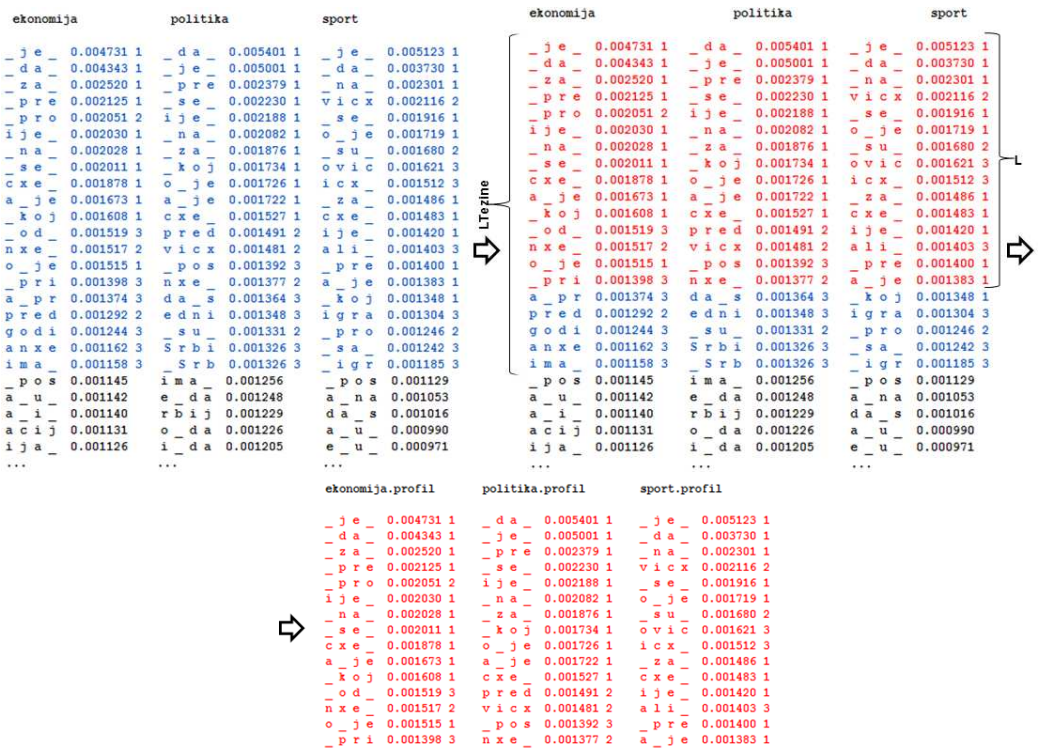


Slika 4.2: Uvođenje težinskih faktora na primeru klasa Ebart-3 korpusa, za vrednosti parametara $n = 4$ i $LTezine = 20$.

- Ukloniti sve n-grame koji ne pripadaju grupi od $LTezine$ najfrekventnijih n-grama.
- Od $LTezine$ najfrekventnijih n-grama ukloniti sve n-grame kojima je dodeljena težina manja od neke unapred zadate vrednosti $PragT$. Tako će broj n-grama biti znatno smanjen.
- Izabrati dužinu profila L, odnosno broj najfrekventnijih n-grama koji će biti razmatrani i koji će određivati profil. Na slici 4.4 je prikazana ilustracija ovog postupka.

Iz prikazanog se može zaključiti da će profil dokumenta za testiranje, s obzirom da se konstruiše na isti način kao u osnovnoj varijanti metode, činiti uređeni skup parova (x_1, f_1), (x_2, f_2),..., (x_L, f_L) L najfrekventnijih n-grama x_i i njihovih normalizovanih frekvencija f_i. U slučaju modifikacije metode na nivou mere različitosti, profil klase će činiti uređeni skup trojki (x_1, f_1, t_1), (x_2, f_2, t_2),..., (x_L, f_L, t_L) L najfrekventnijih n-grama x_i, njihovih normalizovanih frekvencija f_i i njima dodeljenih težinskih faktora t_i. U slučaju modifikacije na niovu profila klase, profil klase će činiti uređeni skup parova (x_1, f_1), (x_2, f_2),..., (x_L, f_L) L n-grama x_i kojima je dodeljena težina t_i veća od neke unapred zadate vrednosti $PragT$ i njihovih normalizovanih frekvencija f_i.

Nad ovako definisanim profilima klasa i dokumenata za testiranje, primenjuje se mera različitosti (modifikovana ili ne) kao kvantitativna odrednica sličnosti, odnosno različitosti između dva profila. Dokument se zatim



Slika 4.3: Konstrukcija profila klase u slučaju modifikacije na nivou mere različitosti na primeru klasa Ebart-3 korpusa, za vrednosti parametara $n = 4$, $L = 15$ i $LTezine = 20$.

pridružuje onoj klasi sa kojom je najsličniji, odnosno ima najmanju vrednost mere različitosti.

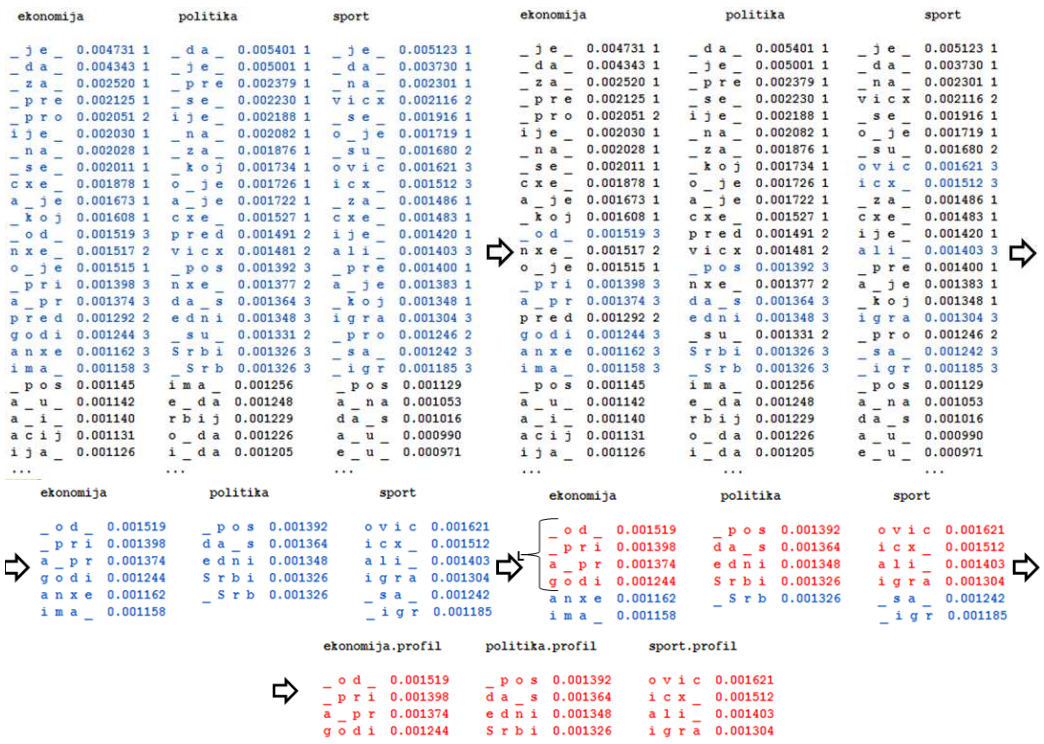
4.2 Metoda zasnovana na wordnet-u

Pitanje koje se postavlja jeste da li je nekako informacije sadržane u leksičkim resursima moguće iskoristiti u cilju bolje klasifikacije dokumenata, odnosno da li uključivanje semantičkih informacija može unaprediti zadatke klasifikacije dokumenata i istraživanja podataka.

Jedan od najpoznatijih radova na tu temu svakako je [67].

Sam postupak klasifikacije opisane u ovom radu sastoji se iz tri prolaza kroz korpus:

1. U prvom prolazu, svim rečima u dokumentu iz posmatranog korpusa pridružuje se oznaka vrste reči (imenice, pridevi, glagoli i drugo).



Slika 4.4: Konstrukcija profila klase u slučaju modifikacije na nivou profila klase na primeru klasa Ebart-3 korpusa, za vrednosti parametara $n = 4$, $L = 15$, $LTezine = 20$ i $PragT = 2$.

2. U drugom prolazu, za svaku imenicu ili glagol, vrši se pregled wordnet-a i pravi se globalna lista svih sinonima i hipernima svake od tih reči. Oni koncepti koji se retko javljaju u korpusu isključuju se iz posmatranja a oni preostali formiraju skup osobina.

3. Tokom trećeg prolaza, izračunava se gustina svakog koncepta (definisana kao odnos broja pojave tog koncepta i ukupnog broja reči u dokumentu).

U ovom radu se definiše i parametar h koji predstavlja veličinu generalizacije odnosno koliko nivoa naviše treba posmatrati hipernime za dati koncept.

Pripadnost nekoj klasi se definiše korišćenjem dobijenih gustina za koncepte. Tako je u ovom radu dat primer sa dve klase, $Istorija$ i $Porez$. Pravilo pripadnosti klasi se definiše preko gustine koncepta "vlasništvo". Ako je njegova gustina mala u nekom dokumentu, onda taj dokument pripada klasi $Istorija$ a u suprotnom pripada klasi $Porez$. Smatra se da se reči

koje se odnose na reč "vlasništvo", uglavnom koriste u tekstovima koji se bave temom poreza i vrlo retko su to neki istorijski dokumenti. Pokazalo se da je ovaj pristup dobar kod tekstova koji koriste nestandardni i prošireni vokabular.

Poznati rad iz ove oblasti je i [63]. U okviru ovog rada koristi se wordnet za unapređivanje metoda zasnovanih na neuronskim mrežama i primenjuje se nad Reuters-21578 novinskom kolekcijom. Rad u kome se takođe istražuje uticaj semantičkih informacija na zadatke klasifikacije tekstova i istraživanja podataka je [65]. U okviru ova dva rada, wordnet se koristi samo u cilju dobijanja sinonima neke reči.

Imajući u vidu objavljene rezultate za klasifikaciju dokumenata na engleskom jeziku, ideja je da se bogati leksički resursi na srspkom jeziku, kao sto su srpski wordnet [43] i Srpski elektronski recnik (Multext-East) [40] (koji su detaljno opisani u poglavlju 2) iskoriste za klasifikaciju teksta na srpskom jeziku na jedan nov i originalan način. U tu svrhu korišćena je xml reprezentacija Ebart-3 korpusa, detaljno opisanog u odeljku 1.2.3.

4.2.1 Procedura klasifikacije

Postupak klasifikacije se može podeliti u dve faze: *učenje*, kada se na osnovu skupa za učenje kreira klasifikator i *testiranje*, kada se na osnovu skupa za testiranje proverava kvalitet napravljenog modela.

U Ebart-3 korpusu dokumenti su podeljeni u tri klase: *Ekonomija*, *Politika* i *Sport*. Neka su u skupu za učenje te klase i dokumenti u njima označeni na sledeći način: $E = \{d_1, d_2, ..., d_{N_E}\}$, $P = \{d_1, d_2, ..., d_{N_P}\}$ i $S = \{d_1, d_2, ..., d_{N_S}\}$, pri čemu $N_E = 333$, $N_P = 935$ i $N_S = 977$ u konkretnom slučaju predstavljaju broj dokumenata pridruženih tim klasama, redom.

Faza učenja kod metode zasnovane na wordnet-u može da se opiše kroz sledeći niz koraka:

- Za svaku klasu iz skupa za učenje formira se lista reči u osnovnom obliku, opadajuće uređena prema frekvenciji pojavljivanja u toj klasi. Pod frekvencijom pojavljivanja reči u klasi podrazumeva se broj pojava svih oblika te reči, u svim dokumentima skupa za učenje te klase. Tako na primer, za reč "sport" svi oblici te reči koji se mogu naći u Srpskom elektronskom rečniku su: "sport", "sporta", "sportu", "sporte", "sportom", "sportovi", "sportova", "sportovima", "sportove". Smatra se da se reč "sport" pojavila 90 puta u klasi *Sport* ako su se svi oblici ove reči pojavli ukupno 90 puta u svim dokumentima za učenje klase *Sport*.

- Ovako dobijenim rečima se dodeljuje oznaka za vrstu reči: imenica, glagol ili neka od preostalih osam vrsta reči u srpskom jeziku (zamenica, pridev, broj, predlog, prilog, uzvik, rečca ili veznik).

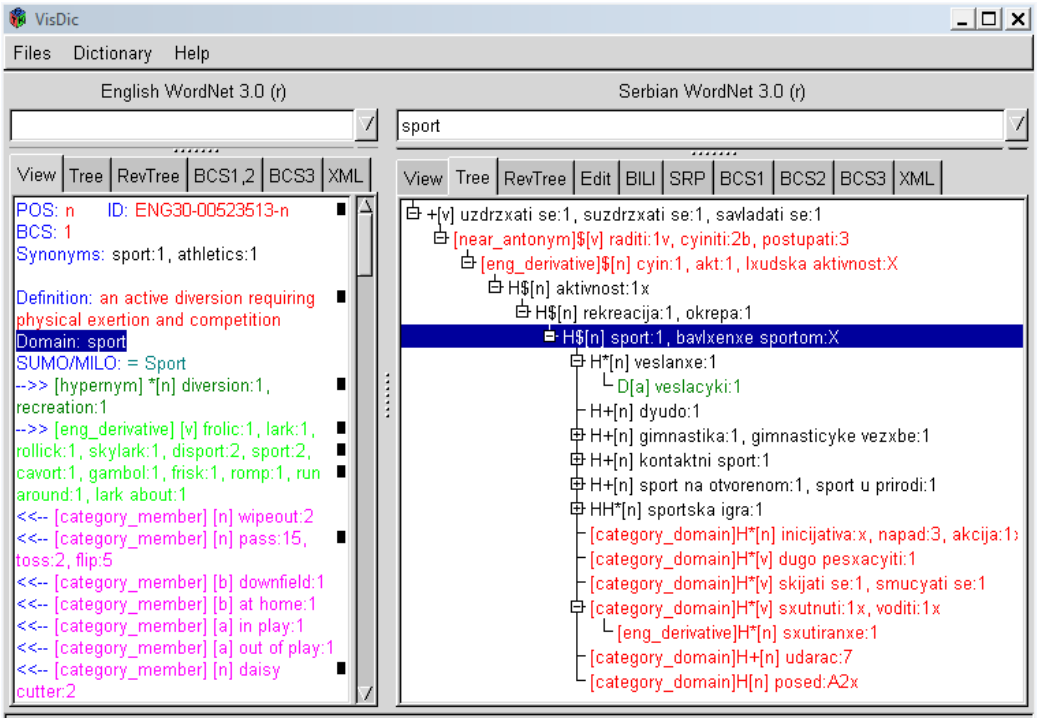
- Iz liste se izbacuju sve reči koje nisu imenice ili glagoli.

- Na osnovu dobijene liste reči, za svaku klasu se formira globalna lista koncepata iz srpskog wordnet-a koja će predstavljati tu klasu. Ova lista se formira tako što se za reči koje su po svojoj učestalosti i specifičnosti važne za posmatranu klasu, u srpskom wordnet-u pronalaze koncepti koji će obuhvatiti jednu ili više takvih reči. Za neku reč se kaže da je obuhvaćena konceptom ako je jednaka nekom od literala sinonima pridruženih tom konceptu ili konceptima koji sa njim grade neku od semantičkih ili leksičkih relacija. Od semantičkih relacija uzete su u obzir sledeće: podređen-nadređen (eng. hyponym/hypernym), celina-deo (eng. holo_part), celina-član (eng. holo_member), antonimija (eng. near_antonym) i relacija koja povezuje značenje i domen korišćenja koncepta (eng. category_domain). Od leksičkih relacija korišćena je relacija izvođenja (eng. derived, eng_derivative). Ovo su ujedno i najčešće relacije u srpskom wordnet-u.

  Primer za koncept "sport" prikazan je na slici 4.5. Ovom konceptu su pridruženi literali sinonimi "sport" i "bavlxenxe sportom". Sa slike se vidi da je koncept "sport" nadređen konceptima "veslanxe", "dyudo", "gimnastika", "kontaktni sport", "sport na otvorenom" i "sportska igra" (gradi relaciju "hypernym" sa njima). Relaciju "category_domain" gradi sa konceptima "inicijativa", "dugo pesxacyiti", "skijati se", "sxutnuti", "udarac" i "posed". Koncept "veslanje" gradi relaciju "derived" sa konceptom "veslacyki" a koncept "sxutnuti" gradi relaciju "eng_derivate" sa konceptom "sxutiranxe". Dakle, kada se kaže da je reč obuhvaćena konceptom "sport", to znači da je jednaka nekom od literala sinonima pridruženih tom konceptu ili konceptima koji grade spomenute veze sa njim. U obzir se uzimaju i koncepti koji posredno grade neku od spomenutih veza sa datim konceptom. Na primer, "sport" gradi vezu "hypernym" sa konceptom "veslanxe" a "veslanxe" gradi vezu "derived" sa konceptom "veslacyki". Dakle, i literali sinonimi pridruženi konceptu "veslacyki" se uzimaju u obzir.

  Ovako dobijeni literali mogu eventualno da se filtriraju po nekom domenu. Svi ovi literali se dodatno filtriraju tako što se iz razmatranja isključuju oni literali koji se ne pojavljuju ni u jednom dokumentu korpusa. Zbog toga se uvodi pojam *aktivnih literala*, odnosno literala koji se u nekom od svojih gramatičkih oblika pojavljuju u bar jednom

dokumentu u čitavom korpusu. Tako će listu pridruženu klasi činiti koncepti, odnosno na gore opisan način njima pridruženi aktivni literali (eventualno filtrirani po nekom domenu).



Slika 4.5: Prikaz koncepta "sport" u srpskom i engleskom wordnet-u korišćenjem VisDic alata.

- U cilju pravilnog izbora koncepata koji će činiti listu predstavnika klase, za svaki koncept koji je kandidat da bude član liste izračunava se težina koja određuje koliko je taj koncept (zajedno sa njemu pridruženim drugim konceptima, odnosno literalima) značajan za datu klasu. Težina koncepta se izračunava na sledeći način:

  Neka je dat koncept k_S koji je kandidat za listu predstavnika klase *Sport*. Težina ovog koncepta se izračunava po sledećoj formuli:

$$tezina(k_S) = tf(k_S, S) * idf(k_S) \tag{4.24}$$

pri čemu je

$$tf(k_S, S) = SrednjaGustinaPoLiteralu(k_S, S)$$
$$idf(k_S) = log(\frac{N}{df_{k_S}} + 0.01) \tag{4.25}$$

gde je N ukupan broj svih dokumenata u korpusu Ebart-3 a df_{k_S} je broj dokumenata u korpusu Ebart-3 u kojima se pojavljuje bar jedan literal (u nekom od svojih oblika) pridružen konceptu k_S.

Neka je konceptu k_S pridruženo M_S aktivnih literala (literala koji se u nekom od svojih oblika pojavljuju u bar jednom dokumentu korpusa) iz srpskog wordnet-a:

$$k_S = \{l_1, l_2, ..., l_{M_S}\} \tag{4.26}$$

Tada je

$$tf(k_S, S) = SrednjaGustinaPoLiteralu(k_S, S) = \\ = \frac{\sum_{i=1}^{M_S} SrednjaGustina(l_i, S)}{M_S} \tag{4.27}$$

pri čemu je

$$SrednjaGustina(l_i, S) = \frac{\sum_{j=1}^{N_S} Gustina(l_i, d_j)}{N_S} \tag{4.28}$$

gde $Gustina(l_i, d_j)$ predstavlja odnos frekvencije ili broja pojave literala l_i u dokumentu d_j i ukupnog broja reči u tom dokumentu. N_S je broj dokumenata koji pripadaju klasi Sport u skupu za učenje.

- Na osnovu težine dodeljene konceptu, određuje se da li će taj koncept biti član liste koja određuje klasu, ili neće. Ukoliko je težina veća od nekog unapred zadatog broja, koncept se pridružuje listi. U suprotnom, pokušava se sa nekim drugim konceptom kandidatom. Za svaku klasu se bira po desetak koncepata koji će biti pridruženi toj klasi. Ovim je faza učenja klasifikatora završena.

Faza testiranja se može opisati kroz sledeći niz koraka:

- Za svaki dokument za testiranje i za svaku klasu, izračunava se mera pripadnosti tog dokumenta toj klasi. Mera pripadnosti dokumenta klasi

izračunava se kao gustina pojavljivanja svih aktivnih literala (eventualno filtriranih po nekom domenu) pridruženih svim konceptima u listi koja određuje tu klasu.

- Dokument se pridružuje onoj klasi za koju ima najveću vrednost mere pripadnosti klasi.

Formalno, mera pripadnosti klasi može da se definiše na sledeći način: Neka je klasi *Ekonomija* pridružena lista koncepata $E = (k_{E_1}, k_{E_2}, \ ..., k_{E_n})$, klasi *Politika* lista $P = (k_{P_1}, k_{P_2}, \ ..., k_{P_n})$ a klasi *Sport* lista $S = (k_{S_1}, k_{S_2}, \ ..., k_{S_n})$. Mera pripadnosti dokumenta, na primer, klasi *Sport* se računa po sledećoj formuli (slično važi i za druge klase):

$$MeraPripadnosti(d, Sport) = \sum_{k_S \in S} \sum_{l \in k_S} Gustina(l, d) \tag{4.29}$$

pri čemu je $Gustina(l, d)$ odnos frekvencije literala l u dokumentu d i ukupnog broja reči u tom dokumentu. Pod frekvencijoom literala l u dokumentu d podrazumeva se broj pojave tog literala i svih njegovih oblika u tom dokumentu. Literali koji se pridružuju konceptima koji čine listu predstavnika neke klase, kao što je već rečeno, mogu da se filtriraju po nekom domenu.

S obzirom na značaj engleskog wordnet-a, njegova struktura je u više navrata proširivana dodatnim informacijama koje bi ga mogle učiniti primenljivijim u prirodnojezičkim obradama. Prvo proširenje odnosi se na proširivanje engleskog wordnet-a semantičkim domenima. Semantički domeni predstavljaju prirodan način da se uspostave semantičke relacije između značenja reči koje bi se mogle uspešno koristiti u raznim domenima obrade prirodnih jezika (naročito u rešavanju problema klasifikacije). Skoro svaki koncept je anotiran bar jednim obeležjem domena koje je izabrano iz skupa od oko dvesta hijerarhijski organizovanih domena [41]. Srpski wordnet još uvek nije proširen domenima ali se oni lako mogu dobiti iz engleskog wordnet-a. Kod (većine) koncepata u srpskom wordnet-u postoji direktna veza sa odgovarajućim konceptima u engleskom wordnet-u. Primer za koncept "sport" dat je na slici 4.5. Na levoj strani slike može da se vidi koncept u engleskom wordnet-u koji odgovara konceptu "sport" u srpskom wordnet-u (desna strana slike). Tom konceptu u engleskom wordnet-u pridruženo je obeležje domena "sport" pa se smatra da je taj domen pridružen i odgovarajućem konceptu u srpskom wordnet-u.

Domeni koji su od interesa u slučaju klasifikacije dokumenata na klase *Ekonomija*, *Politika* i *Sport* su:[1]

[1]Spisak svih domena dostupan je na *http://wndomains.fbk.eu/hierarchy.html*

- *Ekonomija*: economy (banking, enterprise, money, tax), commerce, industy.
- *Politika*: politics, anthropology.
- *Sport*: sport (badminton, baseball, basketball, football, golf, soccer, tennis, volleyball, skiing, rowing, swimming, diving, athletics, boxing, fishing, hunting, bowling).

Za dobijanje domena i literala pridruženih konceptu korišćena je eXist XML baza podataka. Za pregled wordnet-a korišćen je alat VisDic a za izračunavanje težina koncepata i samu proceduru klasifikacije, razvijen je nov alat pod nazivom *WordNetKlasifikacija* koji je implementiran u C programskom jeziku.

5. Rezultati

Metode koje su predstavljene u prethodnom odeljku, testirane su prvenstveno na korpusu tekstova na srpskom jeziku. U tu svrhu korišćen je korpus novinskih članaka iz dnevnog lista "Politika" Ebart-3, predstavljen u odeljku 1.2.3.

5.1 Metoda zasnovana na n-gramima

Kao što je rečeno u odeljku 4.1.1, n-grami mogu biti definisani na nivou bajta, karaktera ili reči. U ovom radu prednost se daje metodi zasnovanoj na n-gramima bajtova. Sve mere različitosti predstavljene u prethodnom odeljku, testirane su na korpusu Ebart-3 za vrednosti parametra $n = 5$, 6 i 7 i parametra $L = 20000$, 40000 i 60000. Pretpostavka je da će testiranje metode za ove vrednosti parametara biti dovoljno u odlučivanju da li je i koliko je neka mera različitosti primenjiva na ovom korpusu. Rezultati testiranja svih pomenutih mera prikazani su u tabelama 5.1 i 5.2. Tačnost dobijene klasifikacije izražena je u terminima mikro- i makro-prosečne F-mere.

Crvenom bojom su označene mere koje daju najbolje rezultate i koje su izabrane za razmatranje u daljem radu. To su: d_1 – mera predstavljena u radu Kešelja, d_2, d_{11} i d_{12} – mere predstavljene u radu Tomovića kao i mere dRP i dSR. O ovim merama bilo je reči u odeljku 4.1.3. Prve četiri mere biće označene sa dK, dT_1, dT_2 i dT_3, redom. Sve ove mere biće detaljno testirane u daljem radu.

5.1.1 N-grami na nivou bajta

Pri testiranju metode zasnovane na n-gramima, prvo pitanje koje se postavlja jeste koje su vrednosti parametara n i L za koje se dobijaju najbolji rezultati, odnosno najveća tačnost metode. U cilju dobijanja odgovora na to pitanje, mere različitosti su testirane za sve vrednosti n i L za koje to ima smisla. Za vrednosti parametra n uzete su vrednosti od 4 do 8. Za sve ostale vrednosti

	$L = 20000$			$L = 40000$			$L = 60000$		
	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$
d_1	95.09	95.27	95.36	95.00	95.00	95.63	95.00	94.92	95.27
d_2	95.18	95.45	95.54	94.83	95.09	95.54	94.83	94.56	95.09
d_3	92.77	93.13	93.31	93.22	93.22	93.31	93.22	93.31	93.31
d_4	58.61	50.58	47.46	62.09	49.60	47.46	66.99	49.96	46.21
d_5	33.10	18.89	12.64	53.70	19.89	14.88	54.06	19.89	14.88
d_6	16.41	06.24	02.85	29.70	12.04	06.59	36.31	14.27	10.87
d_7	18.29	14.27	03.38	31.22	12.49	07.39	38.00	14.54	11.14
d_8	05.00	00.80	00.62	14.01	03.75	00.98	18.91	07.14	02.14
d_9	58.61	07.14	47.46	62.00	49.60	46.21	18.91	49.96	46.21
d_{10}	58.61	50.58	47.46	66.99	49.60	46.21	66.99	49.96	46.21
d_{11}	95.18	95.27	95.45	95.18	95.18	95.45	94.65	94.65	95.18
d_{12}	95.36	95.45	95.63	95.00	95.00	95.18	94.65	94.65	94.92
d_{13}	02.50	00.98	00.54	04.63	01.16	00.80	06.86	01.25	00.45
d_{14}	92.77	93.04	93.13	93.22	93.22	93.31	93.22	93.22	93.31
d_{15}	58.61	50.58	47.46	62.00	49.60	46.21	66.99	49.96	46.21
d_{16}	90.19	90.81	91.17	90.19	90.90	91.44	90.10	90.99	91.44
d_{17}	90.19	90.81	91.17	90.19	90.90	91.44	90.19	90.99	91.44
d_{18}	92.77	93.31	93.31	92.77	93.31	93.67	92.77	93.31	93.67
d_{19}	91.26	92.69	93.76	91.26	92.69	93.76	91.26	92.69	93.76
dRP	95.36	95.36	95.63	95.63	95.90	95.54	95.18	95.09	94.47
dSR	95.42	95.28	95.45	95.54	95.23	95.45	94.30	94.48	95.09

Tabela 5.1: Mikro-prosečna F-mera za različite mere različitosti metode zasnovane na n-gramima bajtova, testirane na korpusu Ebart-3.

parametra n dobijaju se znatno lošiji rezultati, pa one nisu uzete u razmatranje. U cilju ispitivanja ponašanja metode za različite vrednosti parametra L, izvršeno je testiranje mere dK (slično ponašanje pokazuju i druge mere različitosti), za vrednost parametra $n = 6$ i vrednosti parametra L između 5000 i 200000 sa korakom 5000. Na slici 5.1 su prikazani dobijeni rezultati.

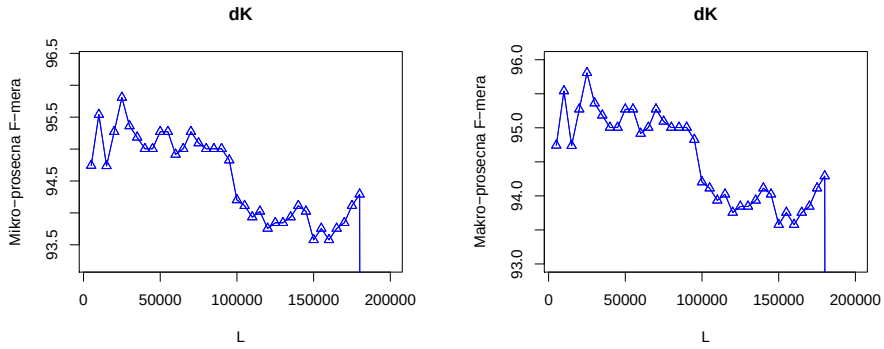
Sa ove slike može da se zaključi da se najveća tačnost metode dobija kada L uzima vrednosti oko 30000. Zbog toga će u daljem radu biti razmatrane vrednosti parametra L u intervalu od 10000 do 50000. Ono što je posebno interesantno jeste nagli pad tačnosti metode za vrednosti parametra L u intervalu od 180000 do 185000. Ovo može da se objasni činjenicom da vrednost mere različitosti zavisi od dužine profila klase, odnosno broja različitih n-grama u klasi. Što je veća dužina profila klase to je veća vrednost mere rezličitosti. U slučaju kada je maksimalna moguća dužina profila neke klase

	$L = 20000$			$L = 40000$			$L = 60000$		
	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$	$n = 5$	$n = 6$	$n = 7$
d_1	94.22	94.18	94.39	94.28	93.83	94.55	93.98	93.92	94.06
d_2	94.35	94.43	94.45	93.83	93.83	94.12	91.98	93.25	93.76
d_3	91.47	91.81	92.19	93.83	91.88	92.19	91.98	92.00	92.19
d_4	49.10	32.59	NDef	52.93	NDef	NDef	58.21	NDef	NDef
d_5	32.44	17.04	10.49	52.87	17.46	04.69	51.88	19.56	10.54
d_6	15.21	NDef	NDef	29.76	08.73	04.69	36.71	09.96	NDef
d_7	16.97	NDef	NDef	31.22	09.14	NDef	38.46	10.20	NDef
d_8	04.76	NDef	00.60	12.10	03.11	NDef	16.43	05.40	01.80
d_9	04.76	32.59	NDef	52.84	NDef	NDef	58.21	NDef	NDef
d_{10}	49.10	32.59	NDef	52.84	NDef	NDef	58.21	NDef	NDef
d_{11}	94.35	94.29	94.38	93.98	94.00	94.34	94.11	93.48	93.91
d_{12}	94.53	94.43	94.60	93.93	93.79	94.34	93.79	93.38	93.46
d_{13}	02.88	NDef	NDef	05.29	01.23	NDef	07.98	01.34	NDef
d_{14}	91.47	94.43	92.00	91.98	91.93	92.19	93.79	91.93	92.19
d_{15}	49.10	32.59	NDef	52.84	NDef	NDef	58.21	NDef	NDef
d_{16}	88.97	89.36	89.51	88.97	89.43	89.81	88.86	89.54	89.81
d_{17}	88.97	89.36	89.51	88.97	89.43	89.81	88.86	89.54	89.81
d_{18}	91.44	91.82	91.87	91.44	91.82	92.24	88.86	91.82	92.24
d_{19}	90.29	91.56	92.47	90.29	91.49	92.40	90.29	91.49	92.40
dRP	94.28	94.24	94.50	94.33	94.91	94.38	93.87	93.86	92.77
dSR	94.31	94.00	94.43	94.23	93.93	94.72	92.54	93.00	93.70

Tabela 5.2: Makro-prosečna F-mera za različite mere različitosti metode zasnovane na n-gramima bajtova, testirane na korpusu Ebart-3.

(ukupni broj različitih n-grama) manja od vrednosti parametra L, manja je i mera različitosti dokumenta i te klase pa će veliki broj dokumenata pogrešno biti klasifikovan u tu klasu. Na pimer, za vrednost parametra $n = 6$, maksimalna moguća dužina profila klase Ekonomija je 184352, klase Politika je 374267 a klase Sport je 388254. Kada vrednost parametra L premaši dužinu najkraćeg profila (u ovom slučaju to je profil klase Ekonomija), dolazi do naglog pada tačnosti metode. U tabeli 5.3 su prikazane vrednosti za mikro- i makro-prosečnu F-meru kada parametar L uzima vrednosti u okolini dužine profila klase Ekonomija. Iz ovoga može da se izvede zaključak da u slučaju kada parametar L premaši dužinu bar jednog profila klase, ova metoda nije primenjiva.

Grafički prikaz velikog broja eksperimenata izvršenih za mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR, za vrednosti parametra n od 4 do 8 i parame-



Slika 5.1: Mikro- i makro-prosečna F-mera za meru različitosti dK i vrednost parametra $n = 6$ za metodu zasnovanu na n-gramima bajtova, testiranu na Ebart-3 korpusu.

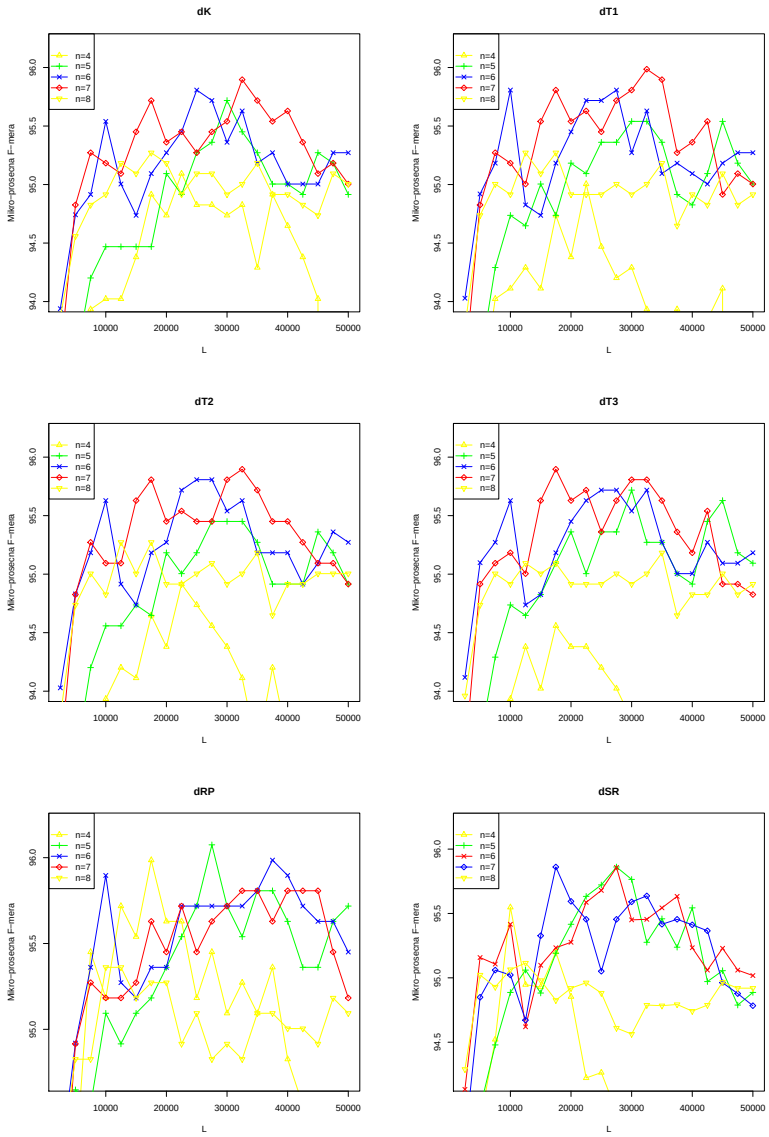
L	$L = 184300$	$L = 184400$	$L = 184500$	$L = 185000$
Mikro F-mera	94.38	84.57	52.72	17.57
Makro F-mera	93.05	81.63	53.38	13.02

Tabela 5.3: Vrednosti za mikro- i makro-prosečnu F-meru, za metodu zasnovanu na n-gramima bajtova primenjenu na Ebart-3 korpusu, za $n = 6$ i za vrednosti parametra L u okolini dužine profila klase Ekonomija.

tra L od 1000 do 50000 sa korakom 1000, prikazan je na slikama 5.2 i 5.3.

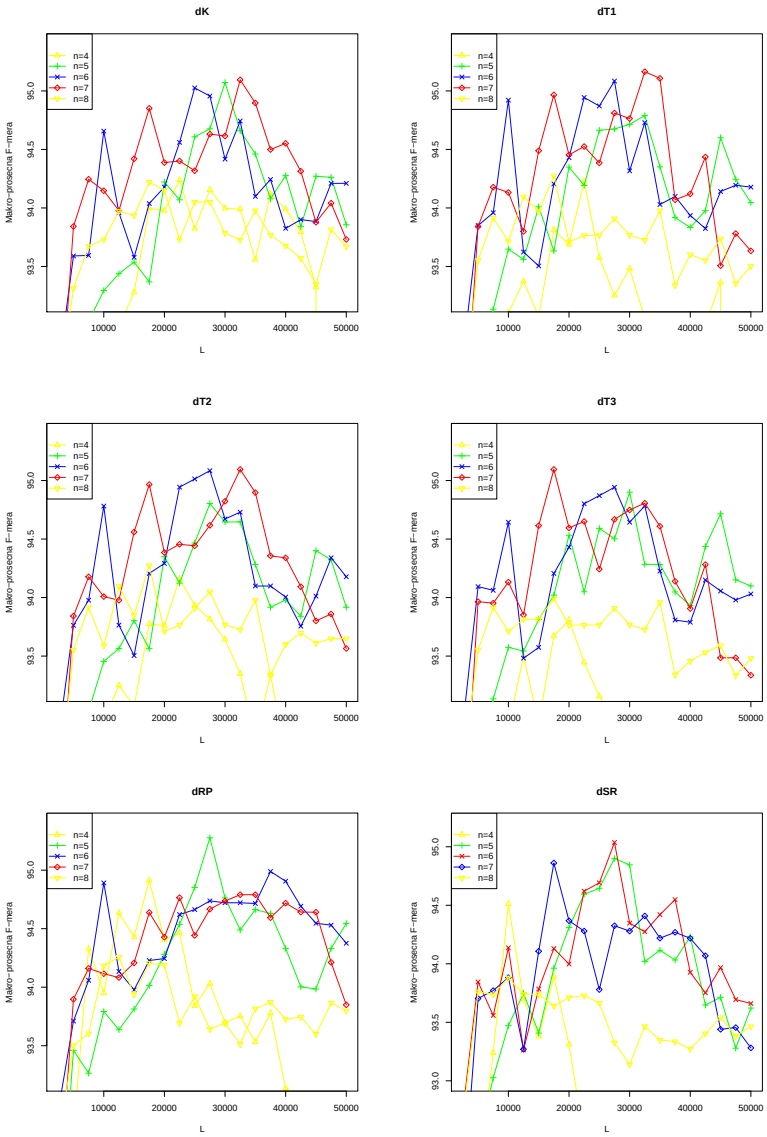
Iz ovog grafičkog prikaza može da se zaključi da ova metoda daje najbolje rezultate kada parametar n uzima vrednosti 5, 6 i 7. Ovo može da se objasni činjenicom da je srednja dužina reči u korišćenom korpusu (a i u srpskom jeziku uopšte), negde oko 5. Ako se iz razmatranja izbace sve reči dužine 1 i 2, srednja dužina reči u korpusu iznosi 6.7 a ako se izbace i reči dužine 3 srednja dužina reči je 7.3. Reči dužine 1, 2 i 3 su obično veznici (i, a, ili, ali, pa, te, ni, tek, već, da, kad,...), predlozi (od, do, iz, iza, pre, uz, na, o, po, pri, u, s, za,...), zamenice (ja, ti, on, ko, ono,...) i slabi su nosioci informacije o sadržaju samog dokumenta.

Na slici 5.4 je prikazano poređenje maksimalnih vrednosti za mikro- i makro-prosečnu F-meru svih razmatranih mera različitosti, bez obzira na vrednosti parametara n i L. Za svaku od mera različitosti, u tabeli 5.4 su prikazane vrednosti tih maksimalnih mikro- i makro-prosečnih F-mera kao i vrednosti parametara n i L za koje se ti maksimumi postižu. Sa slike 5.4 i tabele 5.4 može da se zaključi da se maksimalna vrednost mikro-prosečne F



Slika 5.2: Mikro-prosečna F-mera za mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR metode zasnovane na n-gramima bajtova, testirane na Ebart-3 korpusu.

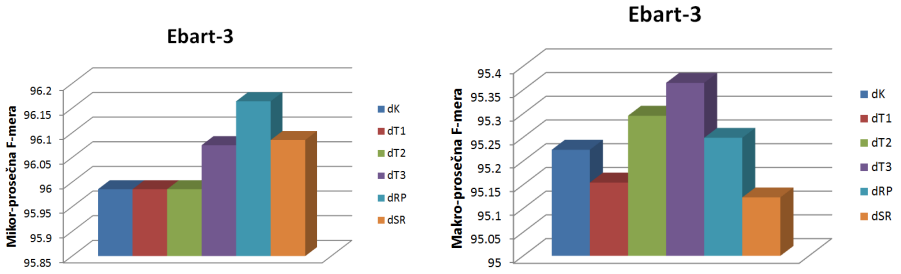
mere postiže za meru različitosti dRP a makro-prosečne F-mere postiže za meru različitosti dT_3.



Slika 5.3: Makro-prosečna F-mera za mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR metode zasnovane na n-gramima bajtova, testirane na Ebart-3 korpusu.

Modifikacija algoritma – težinski faktori

Kao što je već rečeno u prethodnom odeljku, u ovom radu su po prvi put u okviru ove metode uvedeni težinski faktori. Oni su rezultovali modifikacijom algoritma na dva načina: *modifikacija na nivou mere različitosti* i *modi-*



Slika 5.4: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru za mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR metode zasnovane na n-gramima bajtova, testirane na Ebart-3 korpusu.

	Maks mikro-F	n	L	Maks makro-F	n	L
dK	95.9857	7	32000	95.2238	7	32000
dT1	95.9857	7	31000	95.1546	6	27000
dT2	95.9857	6	27000	95.2955	6	27000
dT3	96.0749	6	26000	95.3648	6	26000
dRP	96.1641	6	39000	95.249	6	39000
dSR	96.0854	5	29000	95.1233	5	29000

Tabela 5.4: Vrednosti parametara n i L za maksimalne vrednosti mikro- i makro-prosečne F-mere za mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR metode zasnovane na n-gramima bajtova, testirane na Ebart-3 korpusu.

fikacija na nivou profila klase.

Modifikacija na nivou mere različitosti

Prva vrsta modifikacije algoritma zasniva se na modifikaciji onih mera različitosti koje u okviru izračunavanja uključuju normalizovane frekvencije n-grama. To su mere dK, dT_1, dT_2 i dT_3. Kao što je već rečeno ranije, modifikacija mere se sastoji u tome što se normalizovana frekvencija n-grama koji pripada profilu klase množi težinskim faktorom koji je dodeljen tom n-gramu. U slučaju korpusa Ebart-3, za izračunavanje težinskih faktora korišćena je formula 4.21 prikazana u prethodnom poglavlju:

$$tezina(x) = |C| - c_f + 1$$

gde je $|C|$ ukupan broj klasa u korpusu a c_f broj klasa čiji profil sadrži n-gram x.

Ukoliko se pretpostavi da je $\mathcal{P}_1$ profil klase a $\mathcal{P}_2$ profil dokumenta za testiranje, modifikovane mere različitosti mogu da se prikažu na sledeći način:

$$dKMod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{2 \cdot (f_1(n) * t_1(n) - f_2(n))}{f_1(n) * t_1(n) + f_2(n)} \Big)^2$$

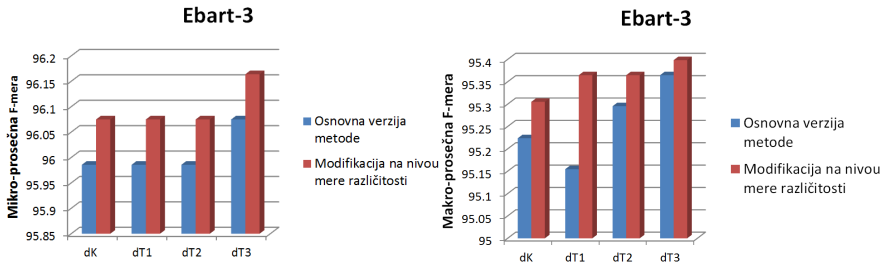
$$dT_1Mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{2|f_1(n) * t_1(n) - f_2(n)|}{f_1(n) * t_1(n) + f_2(n)}$$

$$dT_2Mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \Big(\frac{\sqrt{2}(f_1(n) * t_1(n) - f_2(n))}{\sqrt{(f_1(n) * t_1(n))^2 + f_2(n)^2}} \Big)^2$$

$$dT_3Mod(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{\sqrt{2}|f_1(n) * t_1(n) - f_2(n)|}{\sqrt{(f_1(n) * t_1(n))^2 + f_2(n)^2}}$$

U cilju testiranja efikasnosti ovako definisane modifikacije metode, izvršen je veliki broj eksperimenata. Eksperimenti su izvršeni za vrednosti parametra n od 4 do 8, $LTezine$ od 10000 do 50000 sa korakom 1000 i parametra L od 1000 do $LTezine$ sa korakom 1000. Za sve ostale vrednosti ovih parametara dobijeni su lošiji rezultati pa te vrednosti neće biti uzete u razmatranje.

Na slici 5.5 je prikazano poređenje maksimalnih vrednosti za mikro- i makro-prosečnu F-meru dobijenih primenom svih pomenutih mera različitosti za osnovnu i modifikovanu varijantu metode, bez obzira na vrednosti parametara n, L i $LTezine$. Dodatno, u tabeli 5.5 su prikazane vrednosti parametara n, L i $LTezine$ za koje se ti maksimumi postižu. Može se zaključiti da se za sve mere različitosti dobijaju bolji rezultati primenom modifikovane metode na nivou mere različitosti u odnosu na osnovnu varijantu ove metode.



Slika 5.5: Poređenje osnovne i modifikovane (na nivou mere različitosti) varijante metode u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za metodu zasnovanu na n-gramima bajtova.

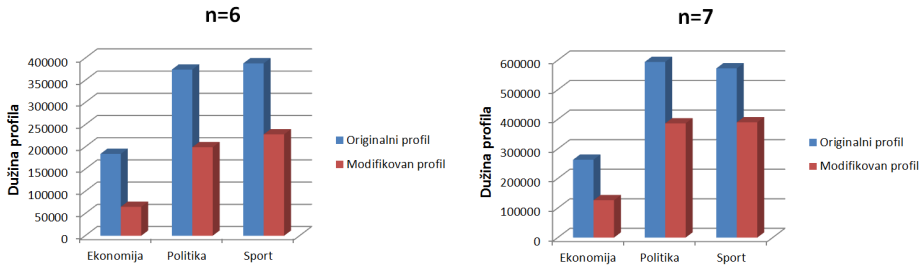
	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	96.0749	6	10000	27000	95.2655	6	10000	27000
dT1	96.07493	6	22000	22000	95.36488	6	22000	22000
dT2	96.07493	6	22000	22000	95.36488	6	22000	22000
dT3	96.1641	7	31000	31000	95.3984	5	29000	31000

Tabela 5.5: Vrednosti parametara n, L i $LTezine$ za koje se dobijaju maksimalne vrednosti mikro- i makro-prosečne F-mere za modifikovanu varijantu metode na nivou mere različitosti zasnovane na n-gramima bajtova, testirane na Ebart-3 korpusu.

Modifikacija na nivou profila klase

Druga vrsta modifikacije algoritma zasnovana je na modifikaciji profila klase. Svaki profil klase čine samo oni n-grami koji se osim u pripadajućoj klasi pojavljuju eventualno u još nekoj od klasa, odnosno samo oni n-grami kojima je dodeljena težina veća od neke unapred zadate vrednosti $PragT$. Tačan postupak konstruisanja profila klase prikazan je u odeljku 4. U slučaju Ebart-3 korpusa, s obzirom na način na koji su definisane težine, parametar $PragT$ može uzimati vrednost 1 ili 2. Eksperimentalni rezultati su pokazali da se bolji rezultati dobijaju za $PragT = 2$. Dakle, profil klase čine samo oni n-grami koji su ekskluzivni za tu klasu, odnosno koji se pojavljuju samo u toj klasi. Nad tako modifikovanim profilima, primenjuje se algoritam iz osnovne varijante metode.

Na slici 5.6 i u tabeli 5.6 dato je poređenje maksimalnih mogućih dužina (ukupan broj različitih n-grama iste dužine) originalnih i modifikovanih profila klasa korpusa Ebart-3, za $n = 6$ i $n = 7$, kada se pri izračunavanju težinskih faktora u razmatranje uzmu svi n-grami iz profila klasa. Može se primetiti značajno smanjenje dužina ovih profila.



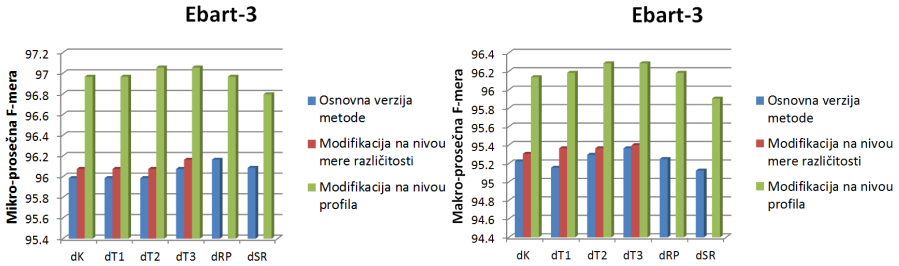
Slika 5.6: Poređenje maksimalnih mogućih dužina originalnih i modifikovanih profila klasa za n-grame na nivou bajta, na Ebart-3 korpusu.

	Dužine originalnih profila		Dužine modifikovanih profila	
	$n = 6$	$n = 7$	$n = 6$	$n = 7$
Ekonomija	184352	263408	65161	126205
Politika	374267	593899	199338	386418
Sport	388254	572378	228359	390084

Tabela 5.6: Maksimalne moguće dužine originalnih i modifikovanih profila klasa za n-grame na nivou bajta, za Ebart-3 korpus.

Za svaku od mera različitosti dK, dT_1, dT_2, dT_3, dRP i dSR, kako bi se proverila efikasnost modifikacije metode na nivou profila, izvršen je veliki broj eksperimenata. Eksperimenti su izvršeni za vrednosti parametra n od 4 do 8, parametra $LTezine$ od 10000 do 50000 sa korakom 1000 i parametra L od 100 do minimalne dužine modifikovanih profila klasa sa korakom 100.

Na slici 5.7 je prikazano poređenje maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za sve mere različitosti za osnovnu varijantu metode i njene modifikacije. U tabeli 5.7 su prikazane vrednosti parametara za koje se dobijaju ti maksimumi. Sa ove slike može da se zaključi da se najbolji rezultati za sve mere različitosti dobijaju za modifikaciju metode na nivou profila klase i to za manje vrednost parametra L u odnosu na osnovnu varijantu metode i njene modifikacije na nivou mere različitosti.



Slika 5.7: Poređenje osnovne i modifikovanih varijanti metode u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za n-grame bajtova, na Ebart-3 korpusu.

Osim na nivou bajta, kao što je ranije rečeno, n-grami mogu biti definisani i na nivou karaktera i reči. Prikazana metoda klasifikacije, osim za n-grame bajtova, može se na potpuno isti način primeniti i na n-grame karaktera i reči.

	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	96.9669	6	7600	26000	96.1359	6	7600	26000
dT1	96.9669	6	5200	25000	96.1832	6	7100	35000
dT2	97.0562	6	5200	25000	96.2864	6	5200	25000
dT3	97.0562	6	5200	25000	96.2864	6	5200	25000
dRP	96.9669	6	7100	35000	96.1832	6	7100	35000
dSR	96.7971	6	5200	25000	95.9042	6	5100	26000

Tabela 5.7: Vrednosti parametara n, L i $LTezine$ za maksimalne vrednosti mikro- i makro-prosečne F-mere za modifikaciju metode na nivou profila, za n-grame bajtova, na Ebart-3 korpusu.

5.1.2 N-grami na nivou karaktera

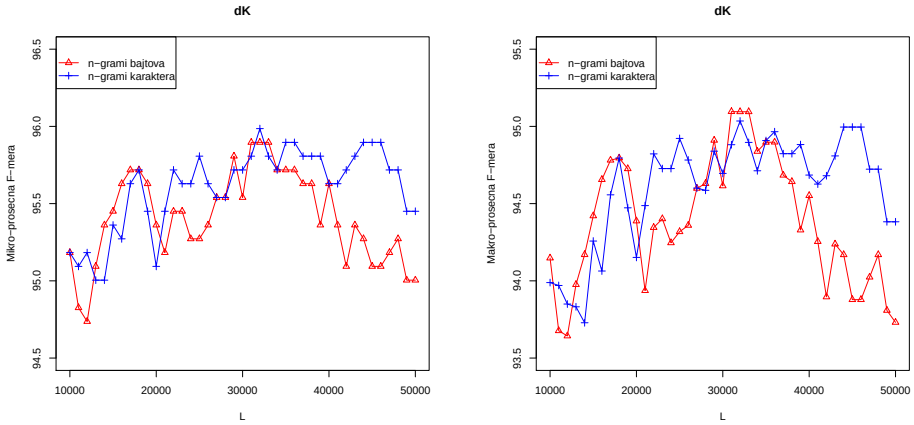
Kako bi poređenje dobijenih rezultata sa rezultatima iste metode zasnovane na n-gramima bajtova bilo što merodavnije, izvršeni su eksperimenti sa istim vrednostima parametara n i L kao u slučaju n-grama bajtova.

Poređenje sa n-gramima bajtova: Na slici 5.8 je prikazano poređenje dobijenih rezultata primenom metode zasnovane na n-gramima karaktera i bajtova, za meru različitosti dK i za vrednost parametra $n = 7$. Slični rezultati dobijeni su i za druge mere različitosti. Poređenje rezultata u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru prikazano je na slici 5.9. Ove maksimalne vrednosti su prikazane u tabeli 5.8 kao i vrednosti parametara n i L za koje se ti maksimumi postižu.

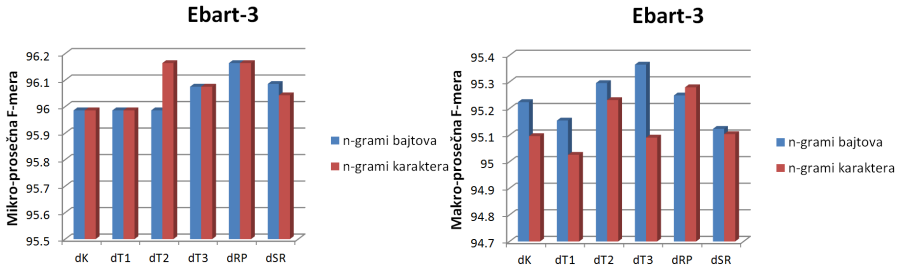
	Maks mikro-F	n	L	Maks makro-F	n	L
dK	95.9857	7	32000	95.0961	6	30000
dT1	96.0749	7	35000	95.089	7	35000
dT2	96.1641	7	35000	95.2314	7	35000
dT3	96.0749	7	34000	95.0903	7	34000
dRP	96.1641	5	24000	95.2797	7	36000
dSR	96.0426	7	33000	95.1034	6	29000

Tabela 5.8: Vrednosti parametara n i L za maksimalne vrednosti mikro- i makro-prosečne F-mere, metode bazirane na n-gramima na nivou karaktera, primenjene na Ebart-3 korpusu.

S obzirom na sličnost između n-grama karaktera i n-grama bajtova, kao



Slika 5.8: Poređenje metode zasnovane na n-gramima bajtova sa istom metodom zasnovanom na n-gramima karaktera, za meru različitosti dK i za vrednost parametra $n = 7$.



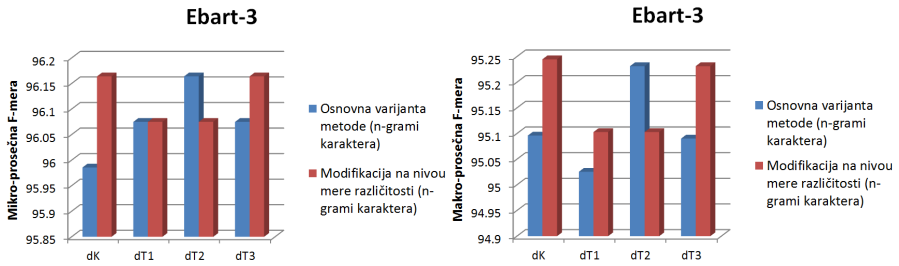
Slika 5.9: Poređenje metode zasnovane na n-gramima bajtova sa metodom zasnovanom na n-gramima karaktera u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

što se moglo i pretpostaviti, dobijeni su veoma slični rezultati. U slučaju korišćenja n-grama karaktera dobijene su iste maksimalne vrednosti za mikro-prosečnu F-meru kao kada se koriste n-grami bajtova, za sve mere različitosti osim za dT_2, gde se dobijaju bolji rezultati i dSR, gde se dobijaju lošiji rezultati. U slučaju maksimalnih vrednosti za makro-prosečnu F-meru, za sve mere različitosti sa izuzetkom dRP dobijaju se bolji rezultati kada se koriste n-grami bajtova. Međutim, sve ove razlike nisu veće od 0.28%

Modifikacija na nivou mere različitosti

Modifikacija na nivou mere različitosti se na potpuno isti način primenjuje u slučaju n-grama karaktera kao i kod n-grama bajtova. Izabrane su iste vrednosti parametara n i L kao kod modifikacije metode za n-grame bajtova.

Poređenje sa osnovnom varijantom metode: Na slici 5.10 je prikazano poređenje maksimalnih tačnosti dobijenih primenom osnovne varijante metode sa njenom modifikacijom na nivou mere različitosti, zasnovane na n-gramima karaktera. Sa slike se može zaključiti da se u slučaju svih mera različitosti, sa izuzetkom mere $dT2$, dobijaju bolji ili isti rezultati primenom modifikacije metode. U tabeli 5.9 su prikazane maksimalne vrednosti za mikro- i makro-prosečnu F-meru kao i vrednosti parametara za koje se ti maksimumi postižu.



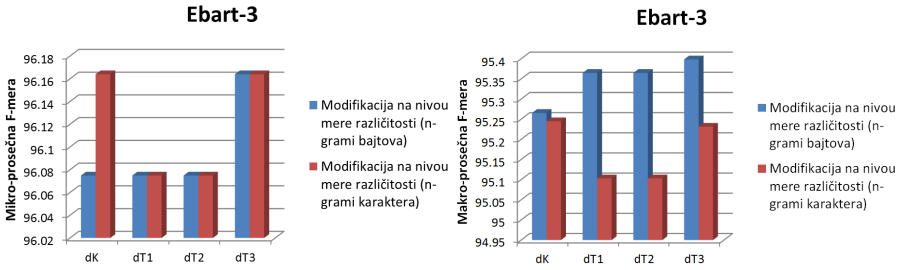
Slika 5.10: Poređenje osnovne varijante metode sa njenom modifikacijom na nivou mere različitosti u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za n-grame karaktera.

	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	96.1641	7	35000	35000	95.2448	7	35000	35000
dT1	96.0749	7	35000	35000	95.103	7	35000	35000
dT2	96.0749	7	35000	35000	95.103	7	35000	35000
dT3	96.1641	7	35000	35000	95.2314	7	35000	35000

Tabela 5.9: Vrednosti parametara n i L za maksimalne vrednosti mikro- i makro-prosečne F-mere, modifikacije na nivou mere različitosti metode zasnovane na n-gramima karaktera, primenjene na Ebart-3 korpusu.

Poređenje sa n-gramima bajtova: Poređenje rezultata dobijenih primenom modifikacije metode na nivou bajta i na nivou karaktera prikazano

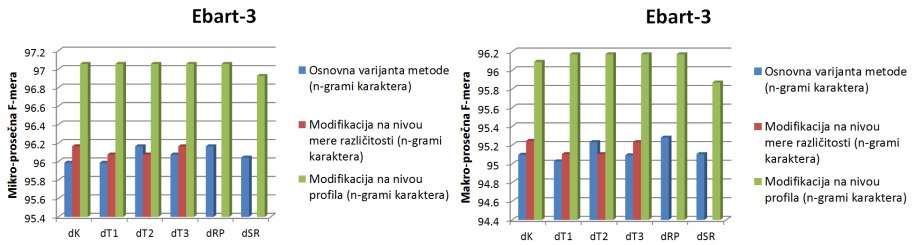
je na slici 5.11. Može se zaključiti da se u slučaju n-grama karaktera za sve mere različitosti dobijaju iste vrednosti za mikro-prosečnu F-meru kao za n-grame bajtova, sa izuzetkom mere dK za koju se dobijaju bolji rezultati. U slučaju makro-prosečne F-mere, dobijaju se lošiji rezultati u odnosu na n-grame bajtova.



Slika 5.11: Poređenje modifikacije na nivou mere različitosti metode zasnovane na n-gramima bajtova sa istom modifikacijom metode zasnovanom na n-gramima karaktera u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

Modifikacija na nivou profila

Poređenje sa osnovnom varijantom metode i njenom modifikacijom na nivou mere različitosti: Poređenje osnovne varijante metode sa njenim modifikacijama, za n-grame karaktera i za sve mere različitosti, prikazano je na slici 5.12. Zaključak je isti kao i u slučaju n-grama bajtova. Najbolji rezultati se dobijaju primenom modifikacije metode na nivou profila.



Slika 5.12: Poređenje osnovne verzije metode sa njenim modifikacijama na nivou mere različitosti i na nivou profila, u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za n-grame karaktera.

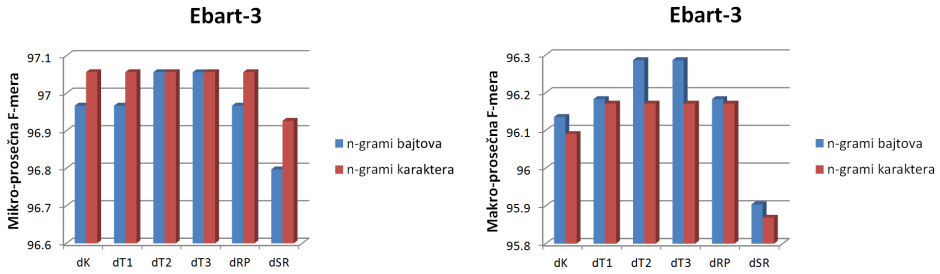
Rezultati u terminima maksimalnih vrednosti za mikro- i makro-prosečnu

F-meru dobijenih primenom modifikacije metode na nivou profila za n-grame karaktera prikazani su u tabeli 5.10.

	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	97.0562	6	8200	32000	96.0905	6	8200	32000
dT1	97.0562	6	4000	24000	96.1713	6	4000	24000
dT2	97.0562	6	4000	24000	96.1713	6	4000	24000
dT3	97.0562	6	4000	24000	96.1713	6	4000	24000
dRP	97.0562	6	4000	24000	96.1713	6	4000	24000
dSR	96.9265	7	3500	26000	95.8689	6	4800	23000

Tabela 5.10: Vrednosti parametara n, L i $LTezine$ za koje se postižu maksimalne vrednosti mikro- i makro-prosečne F-mere za modifikaciju na nivou profila metode zasnovane na n-gramima karaktera, primenjene na Ebart-3 korpusu.

Poređenje sa n-gramima bajtova: Na slici 5.13 je prikazano poređenje modifikacije na nivou profila metode zasnovane na n-gramima bajtova sa istom modifikacijom metode zasnovane na n-gramima karaktera. Može se zaključiti da metoda na nivou n-grama karaktera daje iste ili bolje rezultate za mikro-prosečnu a lošije rezultate za makro-prosečnu F-meru.

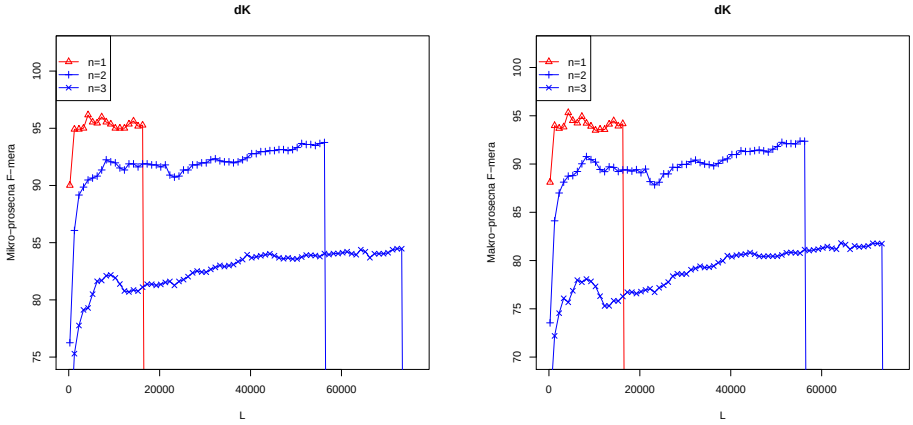


Slika 5.13: Poređenje modifikacije na nivou profila metode zasnovane na n-gramima bajtova sa metodom zasnovanom na n-gramima karaktera u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

5.1.3 N-grami na nivou reči

Za metodu zasnovanu na n-gramima reči, izvršeni su eksperimenti za vrednosti parametara n od 1 do 3 i L od 250 do 75000 sa korakom 250, za sve ranije

pomenute mere različitosti. Na slici 5.14 se mogu videti rezultati dobijeni primenom ove metode za meru različitosti dK i pomenute vrednosti parametara n i L. Slični se rezultati dobijaju i primenom drugih mera različitosti. Može se zaključiti da se najbolji rezultati dobijaju za $n = 1$, odnosno kad se tekst posmatra kao "vreća" od L najfrekventnijih reči. U tabeli 5.11 su prikazane maksimalne moguće dužine profila klasa dobijenih u slučajevima kada parametar n uzima vrednosti 1, 2 i 3. Imajući ovu tabelu u vidu, sa slike se može uočiti nagli pad tačnosti metode kada L premaši dužinu makar jednog profila klase i u tom slučaju ova metoda nije primenjiva. O ovoj pojavi bilo je reči na početku ovog poglavlja.



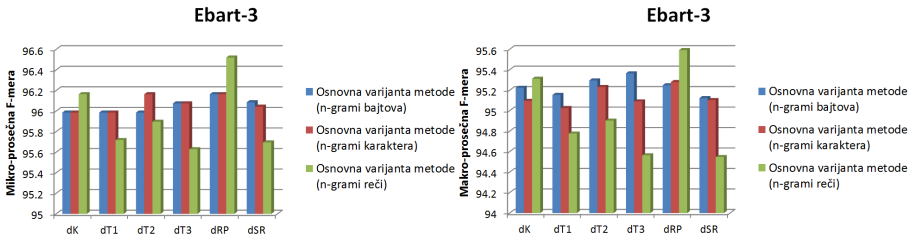
Slika 5.14: Mikro- i makro-prosečna F-mera za meru različitosti dK za metodu baziranu na n-gramima na nivou reči, na Ebart-3 korpusu.

	$n = 1$	$n = 2$	$n = 3$
Ekonomija	16838	56294	73864
Politika	34361	150756	215381
Sport	32079	127407	177910

Tabela 5.11: Maksimalne moguće dužine profila klasa Ebart-3 korpusa, za n-grame na nivou reči.

Poređenje sa n-gramima bajtova i karaktera: Poređenje maksimalnih tačnosti dobijenih primenom osnovne varijante metode zasnovane na n-gramima bajtova, karaktera i reči, prikazano je na slici 5.15. Tačnost je

izražena u terminima mikro- i makro-prosečne F-mere. Vrednosti ovih maksimuma kao i vrednosti parametara n i L za koje se ti maksimumi postižu, prikazane su u tabeli 5.12. Može se zaključiti da se u slučaju mera različitosti dK i dRP najveća tačnost postiže za n-grame reči, dok se za sve ostale mere različitosti postižu lošiji rezultati u odnosu na n-grame bajtova i karaktera.



Slika 5.15: Poređenje osnovne varijante metode zasnovane na n-gramima reči sa istom metodom zasnovanom na n-gramima bajtova i karaktera, u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

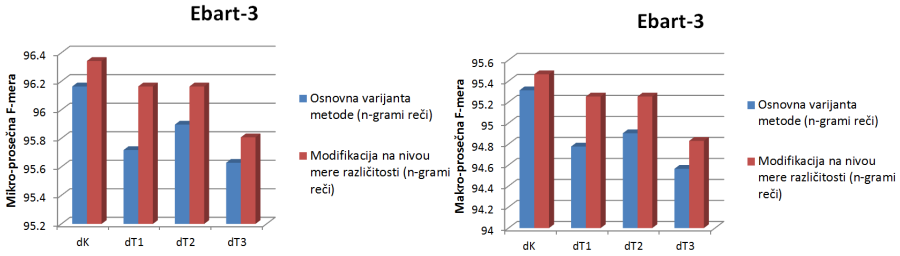
	Maks mikro-F	n	L	Maks makro-F	n	L
dK	96.1641	1	4250	95.3123	1	4250
dT1	95.7181	1	4250	94.7769	1	4250
dT2	95.8965	1	7250	94.9029	1	4250
dT3	95.6289	1	7250	94.5644	1	4250
dRP	96.5209	1	5000	95.5925	1	5000
dSR	95.6946	1	7500	94.5475	1	4000

Tabela 5.12: Vrednosti parametara n i L za maksimalne vrednosti mikro- i makro-prosečne F-mere, metode bazirane na n-gramima na nivou reči, primenjene na Ebart-3 korpusu.

Modifikacija na nivou mere različitosti

U slučaju modifikacije na nivou mere različitosti, izvršeni su eksperimenti za vrednost parametara $n = 1$, $LTezine$ od 1000 do 20000 sa korakom 1000 i L od 100 do $LTezine$ sa korakom 100.

Poređenje sa osnovnom varijantom metode: Na slici 5.16 je prikazano poređenje rezultata dobijenih primenom osnovne varijante ove metode sa njenom modifikacijom na nivou mere različitosti za n-grame na nivou reči. Dodatno, u tabeli 5.13 su prikazane vrednosti parametara n, L i $LTezine$ za koje se ti maksimumi postižu. Sa slike se može uočiti da se za sve mere različitosti povećava tačnost metode nakon ove vrste modifikacije.

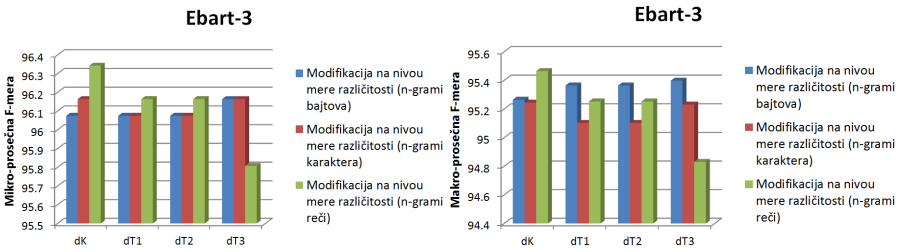


Slika 5.16: Poređenje osnovne varijante metode sa njenom modifikacijom na nivou mere različitosti u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za n-grame reči.

	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	96.3425	1	4300	5000	95.4651	1	4300	6000
dT1	96.1641	1	4000	5000	95.2532	1	4000	5000
dT2	96.1641	1	4000	4000	95.2532	1	4000	4000
dT3	95.8073	1	4300	5000	94.8304	1	4300	5000

Tabela 5.13: Vrednosti parametara n, L i $LTezine$ za koje se postižu maksimalne vrednosti mikro- i makro-prosečne F-mere za modifikaciju na nivou mere različitosti metode zasnovane na n-gramima reči, primenjene na Ebart-3 korpusu.

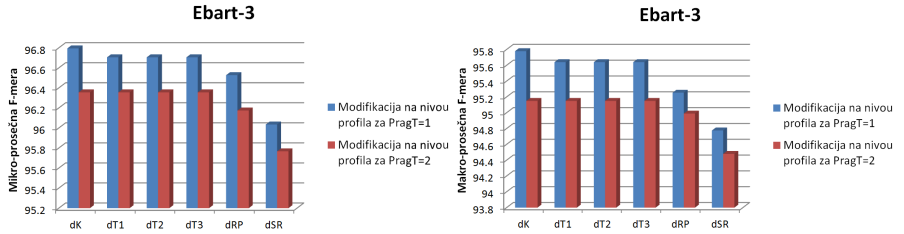
Poređenje sa n-gramima bajtova i karaktera: Poređenje tačnosti modifikacije metode na nivou mere različitosti za n-grame bajtova, karaktera i reči prikazano je na slici 5.17. Sa ove slike može da se zaključi da se najveća tačnost dobija za n-grame reči, za sve mere različitosti sa izuzetkom mere dT_3. U terminima makro-prosečne F-mere, najveća tačnost se postiže za n-grame bajtova, sa izuzetkom mere dK.



Slika 5.17: Poređenje modifikacije na nivou mere različitosti metode zasnovane na n-gramima reči sa istom modifikacijom metode zasnovanom na n-gramima bajtova i karaktera u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

Modifikacija na nivou profila

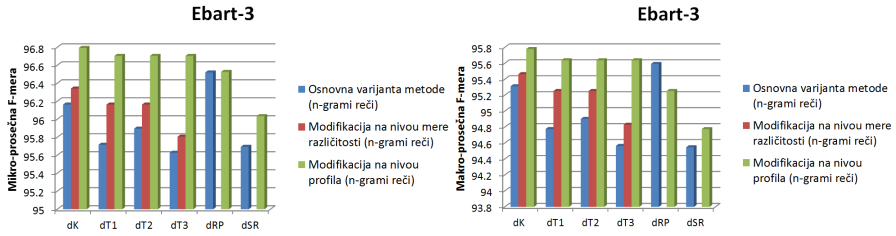
Izvršeni su eksperimenti za vrednosti parametra $n = 1$, $LTezine$ od 1000 do 20000 i L od 100 do minimalne dužine modifikovanog profila sa korakom 100. U slučaju n-grama reči, eksperimentalni rezultati su pokazali da se veća tačnost ove modifikacije metode dobija kada profil klase čine oni n-grami čija je težina veća od $PragT = 1$ a ne od $PragT = 2$ kao u slučaju n-grama karaktera i bajtova. Na slici 5.18 je prikazano poređenje dobijenih rezultata. Zbog toga, u ovom slučaju, profil klase će činiti sve one reči (n-grami za $n = 1$) koje se pojavljuju samo u toj ili eventualno u još jednoj od klasa.



Slika 5.18: Poređenje modifikacije metode na nivou profila za vrednosti $PragT = 1$ i $PragT = 2$, za n-grame reči u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru.

Poređenje sa osnovnom varijantom metode i njenom modifikacijom na nivou mere različitosti: Na slici 5.19 je prikazano poređenje rezultata dobijenih primenom osnovne varijante ove metode sa njenim modifikacijama za n-grame na nivou reči. Tabela 5.14 sadrži informacije o dobijenim maksi-

malnim vrednostima za mikro- i makro-prosečnu F-meru kao i o vrednostima parametara za koje se ti maksimumi postižu.



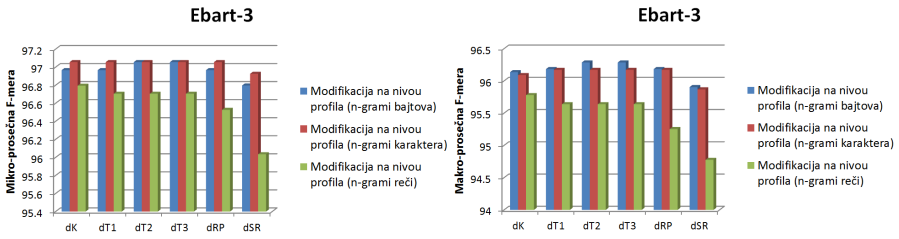
Slika 5.19: Poređenje osnovne varijante metode sa njenim modifikacijama na nivou mere različitosti i na nivou profila u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru za n-grame reči.

	Maks mikro-F	n	L	LTezine	Maks makro-F	n	L	LTezine
dK	96.7943	1	2100	13000	95.7790	1	2100	13000
dT1	96.705	1	2100	13000	95.639	1	2100	13000
dT2	96.705	1	2100	13000	95.639	1	2100	13000
dT3	96.705	1	2100	13000	95.639	1	2100	13000
dRP	96.5271	1	2100	13000	95.2532	1	2100	13000
dSR	96.0636	1	4300	14000	94.7755	1	2700	4000

Tabela 5.14: Vrednosti parametara n, L i $LTezine$ za koje se postižu maksimalne vrednosti mikro- i makro-prosečne F-mere za modifikaciju na nivou profila metode zasnovane na n-gramima reči, primenjene na Ebart-3 korpusu.

Poređenje sa n-gramima bajtova i karaktera: poređenje modifikacije metode na nivou profila zasnovane na n-gramima bajtova, karaktera i reči prikazano je na slici 5.20. Rezultati koji se dobijaju ukazuju na to da se primenom ove modifikacije metode za n-grame reči dobijaju lošiji rezultati u poređenju sa n-gramima karaktera i bajtova.

Iz svega što je do sada prikazano može da se zaključi da metoda i njene modifikacije na nivou n-grama bajtova i karaktera daju slične rezultate. Kada se uporede sa metodom zasnovanom na n-gramima reči, u nekim slučajevima, za neke mere različitosti, metoda zasnovana na n-gramima reči daje bolje rezultate. Međutim, kada se primene modifikacije na nivou mere različitosti a posebno na nivou profila klase, bolji rezultati se dobijaju primenom metode zasnovane na n-gramima bajtova i karaktera.



Slika 5.20: Poređenje modifikacije na nivou profila metode zasnovane na n-gramima reči sa istom modifikacijom metode zasnovanom na n-gramima bajtova i karaktera u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, na Ebart-3 korpusu.

U daljem radu biće razmatrana samo metoda zasnovana na n-gramima bajtova.

5.1.4 Testiranje nezavisnosti metode od jezika

Jedna od glavnih prednosti metode zasnovane na n-gramima bajtova jeste njena potpuna nezavisnost od jezika na kome su napisani dokumenti koji se klasifikuju. U cilju ilustracije ove važne osobine metode, izvršena su mnogobrojna testiranja na korpusima na engleskom, kineskom i arapskom jeziku. Korišćeni su korpusi predstavljeni u okviru poglavlja 1.2.3: 20-Newsgropus i Reuters-21578 na engleskom, Tancorp-12 na kineskom i Mesleh-10 na arapskom jeziku. Testiranje je izvršeno primenom osnovne varijante metode, njene modifikacije na nivou mere različitosti i modifikacije na nivou profila klase, opisanih u prethodnom poglavlju.

U slučaju modifikacije na nivou mere različitosti, pokazalo se da se najbolji rezultati dobijaju kada se težinski faktori računaju po formuli 4.22 prikazanoj u prethodnom poglavlju 4.1.4:

$$tezina(x) = \frac{|C|^2}{c_f^2}$$

pri čemu je $|C|$ ukupan broj klasa u korpusu a c_f broj klasa čiji profil sadrži n-gram x.

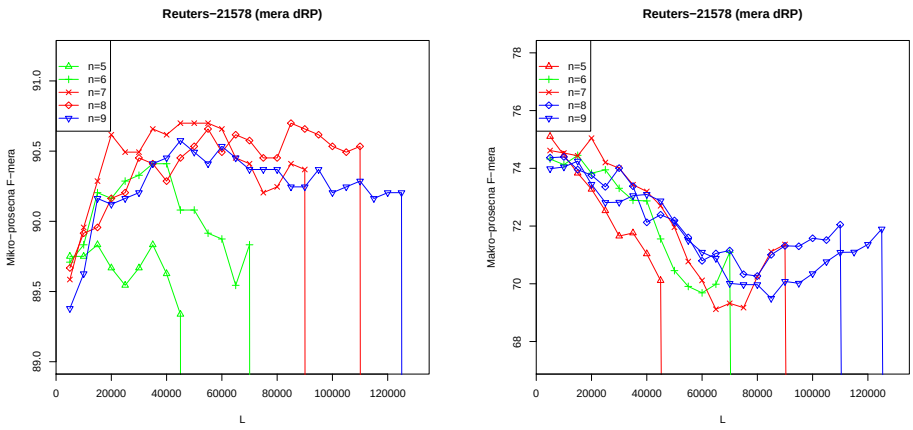
Prilikom izračunavanja težinskih faktora, za parametar $LProf$ izabran je minimum svih mogućih maksimalnih dužina profila klasa za konkretan korpus. Time je postignuto da se pri izračunavanju težina uzima u obzir isti, maksimalan mogući broj n-grama u okviru profila klasa.

Napomenimo još jednom da je modifikacija na nivou mere različitosti definisana samo za mere različitosti koje u svom izračunavanju uključuju

normalne frekvencije n-grama (dK, dT_1, dT_2 i dT_3) a modifikacija na nivou profila klase je definisana za sve ranije pomenute mere različitosti (dK, dT_1, dT_2, dT_3, dRP i dSR). U daljem tekstu prilikom prikazivanja rezultata, sa "OV" će biti označena osnovna varijanta metode, sa "Mod MR" modifikacija na nivou mere različitosti a sa "Mod PK" modifikacija na nivou profila klase.

Reuters-21578

Prvo pitanje na koje treba dati odgovor jeste koje su to vrednosti parametara n i L za koje se dobijaju najbolji rezultati. Prvi skup eksperimenata u slučaju Reuters korpusa, izvršen je za vrednosti parametara n od 5 do 9 i L od 5000 do 130000 sa korakom 5000 za sve ranije pomenute mere različitosti dK, dT_1, dT_2, dT_3, dRP i dSR. Na slici 5.21 su prikazani dobijeni rezulati za meru različitosti dRP, dok su za sve ostale mere različitosti rezultati prikazani na slikama 1 i 2 u poglavlju Prilog 1.

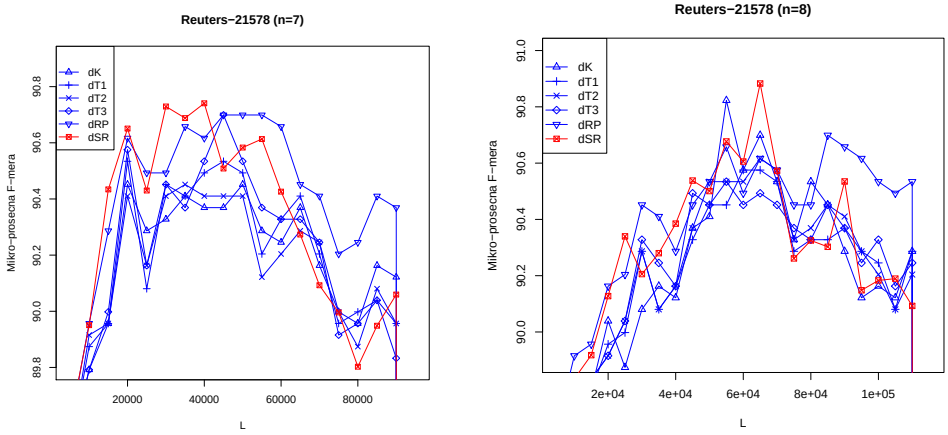


Slika 5.21: Mikro- i makro-prosečna F-mera za Reuters-21578 korpus, za različite vrednosti parametra n i meru različitosti dRP

Kod Reuters-21578 korpusa rezultati su pokazali da se maksimalne vrednosti za mikro-prosečnu F-meru postižu za $n = 7$ i $n = 8$ a makro-prosečnu F-meru za $n = 5$ i $n = 7$, u zavisnosti od primenjene mere različitosti. Primetimo i to da mikro-prosečna F-mera svoj maksimum dostiže za dužinu profila L veću od 40000 a makro-F mera za dužinu profila manju od 25000, za sve posmatrane mere različitosti. Kao i u slučaju Ebart-3 korpusa, na ovim slikama se može uočiti nagli pad performansi metode nakon što L dostigne vrednost veću od maksimalne moguće dužine profila makar jedne

klase u korpusu, i smatra se da tada metoda više nije primenjiva. U slučaju Reuters korpusa, najmanja klasa (klasa sa najmanjim brojem različitih n-grama) je klasa "corn". Za $n = 5$, maksimalna moguća dužina profila ove klase, odnosno ukupan broj svih mogućih različitih 5-grama u ovoj klasi je 49110, za $n = 6$ to je 72105, za $n = 7$ je 93746, za $n = 8$ je 113191, a za $n = 9$ to je 129244. O ovom fenomenu bilo je više reči ranije u okviru poglavlja 5.1.1, kada je ova metoda primenjena na Ebart-3 korpusu.

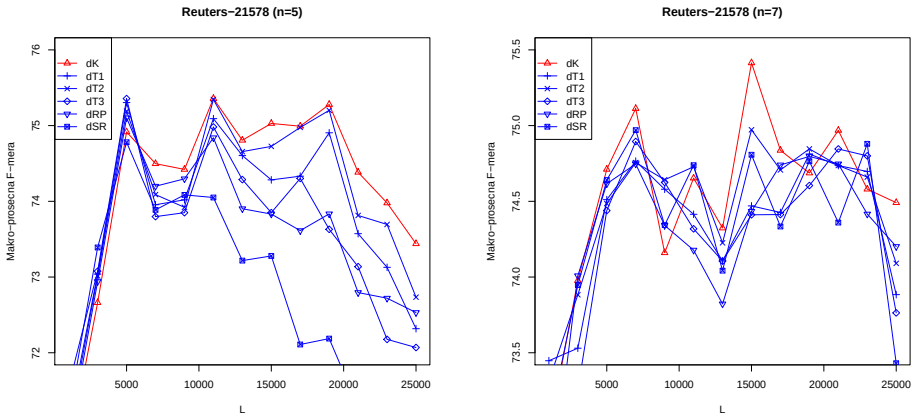
Drugi skup eksperimenata izvršen je u cilju određivanja mere različitosti za koju se dobijaju najbolji rezultati na Reuters korpusu. Za vrednosti dužine n-grama koje su se izdvojile kao karakteristične, izvršeno je poređenje svih mera različitosti. Za mikro-F meru, poređenje je izvršeno za vrednosti parametra $n = 7$ i $n = 8$ a za makro-F meru za $n = 5$ i $n = 7$, za koje se postižu maksimalne tačnosti. Rezultati poređenja za mikro-F meru prikazani su na slici 5.22, za L u intervalu od 5000 do 130000 sa korakom 5000, a za makro-F meru na slici 5.23 za L od 1000 do 25000 sa korakom 2000.



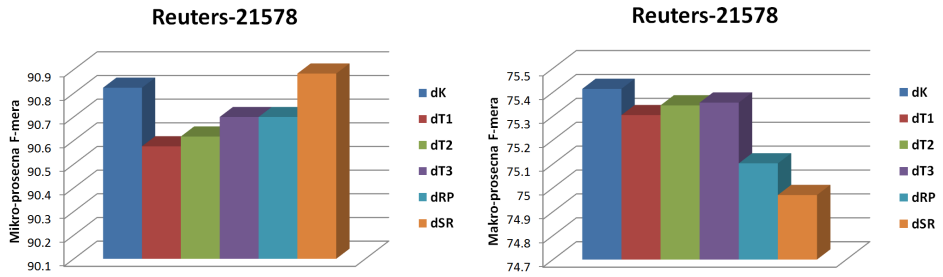
Slika 5.22: Poređenje mera različitosti na Reuters-21578 korpusu, u terminu mikro-prosečne F-mere, za n=7 i n=8.

Na slici 5.24 je prikazano poređenje maksimalnih vrednosti tačnosti za mere dK, dT_1, dT_2, dT_3, dRP i dSR postignute na Reuters korpusu bez obzira na vrednosti parametara n i L. Dodatno, u tabeli 5.15 su prikazane vrednosti ovih parametara za koje se ti maksimumi postižu.

Ono što je posebno karakteristično za Reuters korpus, a što se može zaključiti iz svih do sada prikazanih rezultata, jeste velika razlika između vrednosti za mikro- i makro-prosečnu F-meru. Razlog tome je izrazito neravnomerna raspodela klasa u Reuters korpusu (videti sliku 1.4) i činjenica



Slika 5.23: Poređenje mera različitosti na Reuters-21578 korpusu, u terminu makro-prosečne F-mere, za n=5 i n=7.



Slika 5.24: Poređenje mera različitosti na Reuters-21578 korpusu, u terminu maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, bez obzira na vrednost parametara n i L.

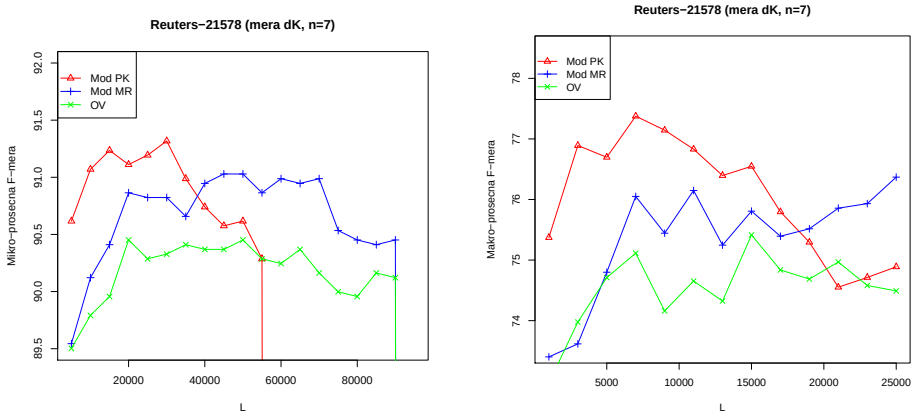
da mikro-F mera (koja daje prednost klasama sa velikim brojem instanci) postiže veoma visoke vrednosti za klase sa veoma velikim brojem dokumenata a niske vrednosti za klase sa veoma malim brojem dokumenata.

Modifikacija na nivou mere različitosti i profila klase Veliki broj eksperimenata je izvršen u cilju poređenja osnovne varijante metode i njenih modifikacija. Eksperimenti su izvršeni za $n = 7$ i $n = 8$ u terminu mikro-F mere, za L od 5000 do 130000 sa korakom 5000, odnosno za $n = 5$ i $n = 7$ u terminu makro-F mere, za L od 1000 do 25000 sa korakom 1000. U slučaju modifikacije na nivou profila klase, pokazalo se da se za $n = 7$ i $n = 8$ najbolji rezultati dobijaju kada se pri generisanju profila klase posmatraju samo oni

OV	*dK*	*dT1*	*dT2*	*dT3*	*dRP*	*dSR*
Mikro F-mera	90.8228	90.5753	90.6166	90.6991	90.6991	**90.8829**
n	8	8	8	7	7	**8**
L	55000	60000	65000	45000	45000	**65000**
Makro F-mera	**75.4143**	75.3044	75.3452	75.3568	75.1025	74.9701
n	**7**	5	5	5	5	7
L	**15000**	5000	5000	5000	5000	7000

Tabela 5.15: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za osnovnu varijantu metode zasnovane na n-gramima bajtova, primenjene na Reuters-21578 korpusu.

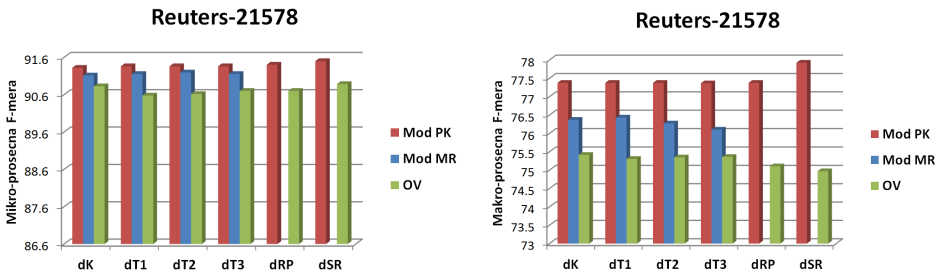
n-grami koji su osim u pripadajućoj klasi sadržani u još eventualno 3 klase a za $n = 5$ u još eventualno 6 od ukupno 10 klasa. Na slici 5.25 je prikazano poređenje osnovne varijante metode sa njenim modifikacijama, za meru dK i $n = 7$. U okviru poglavlja Prilog 1, na slikama 3 i 4 su prikazani rezultati poređenja osnovne varijante metode i njenih modifikacija u terminu mikro-prosečne F-mere za $n = 7$ i $n = 8$ a na slikama 5 i 6 su prikazani rezultati poređenja u terminu makro-prosečne F-mere za $n = 5$ i $n = 7$, za sve mere različitosti.



Slika 5.25: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za $n = 7$ i meru različitosti dK, u terminima mikro- i makro-prosečne F-mere.

Poređenje maksimalnih vrednosti za mikro- i makro-prosečnu F-meru do-

bijenih za svaku meru različitosti pojedinačno, primenom osnovne varijante metode i njenih modifikacija, bez obzira na vrednosti parametara n i L, prikazano je na slici 5.26. U tabelama 5.16 i 5.17 su prikazane i vrednosti parametara n i L za koje se ti maksimumi postižu, u slučaju modifikacije na nivou mere različitosti i modifikacije na nivou profila klase, redom. Vrednosti parametara n i L za koje se postižu maksimumi primenom osnovne varijante metode prikazani su ranije u tabeli 5.15.



Slika 5.26: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za sve mere različitosti u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru, bez obzira na vrednost parametara n i L.

Mod MR	$dKmod$	$dT1mod$	$dT2mod$	$dT3mod$
Mikro F-mera	91.1115	91.1528	**91.194**	91.1528
n	8	8	**8**	8
L	55000	55000	**55000**	55000
Makro F-mera	**76.3693**	76.4304	76.2696	76.104
n	**7**	7	7	7
L	**25000**	15000	15000	7000

Tabela 5.16: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou mere različitosti*, primenjenu na Reuters-21578 korpusu.

Na osnovu dobijenih rezultata može da se zaključi da modifikacija na nivou profila klase povećava tačnost metode (mikro-F mera se povećava sa 90.88 na 91.5, makro-F mera se povećava sa 75.41 na 77.92) a optimalni rezultati se dobijaju za manje vrednosti parametra L u odnosu na osnovnu varijantu metode (u slučaju mikro-F mere L se smanjuje sa 65000 na 35000 a

Mod PK	*dK*	*dT*1	*dT*2	*dT*3	*dRP*	*dSR*
Mikro F-mera	91.3178	91.359	91.359	91.359	91.4002	**91.4968**
n	7	8	7	8	8	**8**
L	30000	35000	30000	35000	25000	**35000**
Makro F-mera	77.376	77.376	77.376	77.3636	77.376	**77.9236**
n	7	5	5	5	5	**7**
L	7000	5000	5000	5000	5000	**7000**

Tabela 5.17: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou profila klase*, primenjenu na Reuters-21578 korpusu.

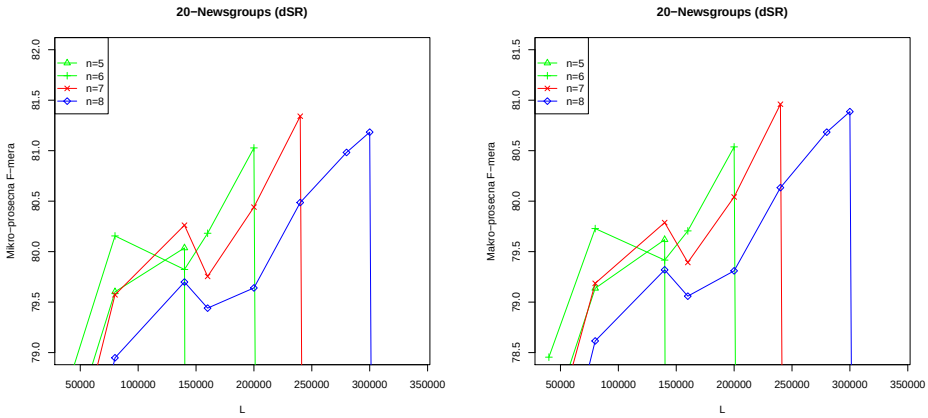
u slučaju makro-F mere smanjuje se sa 15000 na 7000), čime se povećava efikasnost metode. Takođe, prostor pretrage optimalnog rešenja po parametru L se smanjuje s obzirom na to da L može uzimati samo vrednosti manje ili jednake maksimalnom mogućem broju različitih n-grama koji čine profil najmanje klase u korpusu. Nakon što L premaši tu vrednost, tačnost metode naglo pada pa ona više nije primenjiva. U slučaju modifikacije na nivou profila, ovaj maksimalan mogući broj n-grama koji čine profil klase se smanjuje jer se iz razmatranja isključuju svi n-grami koji se pojavljuju u više od unapred definisanog, dozvoljenog broja klasa. Tako na primer, za $n = 5$ maksimalan mogući broj n-grama koji čine profil najmanje klase, odnosno najveća moguća vrednost parametra L, smanjuje se sa 49110 na 31428 (iz profila klase isključuju se svi n-grami koji se pojavljuju u više od 7 klasa), za $n = 7$ sa 93746 na 59198 a u slučaju $n = 8$ sa 113191 na 81484 (isključuju se svi n-grami koji se pojavljuju u više od 4 klase).

20-Newsgroups

Drugi korpus na engleskom jeziku za koji je izvršeno testiranje metode zasnovane na n-gramima bajtova je 20-Newsgroups korpus. To je korpus koji se po svojim osobinama po mnogo čemu razlikuje od Reuters korpusa, na primer, ima ravnomernu raspodelu broja dokumenata po klasama i karakteriše se nepreklapajućim klasama (svakom dokumentu se pridružuje po tačno jedna klasa).

U cilju određivanja vrednosti parametara n i L za koje se dobijaju optimalni rezultati na ovom korpusu, osnovna varijanta metode je testirana za vrednosti parametra n od 5 do 8 i za različite vrednosti parametra L u intervalu od 1000 do 300000. Rezultati ovih testiranja za meru dSR prikazani su na slici 5.27. Za ostale mere različitosti rezultati su prikazani na slikama

7 i 8 u okviru poglavlja Prilog 1.

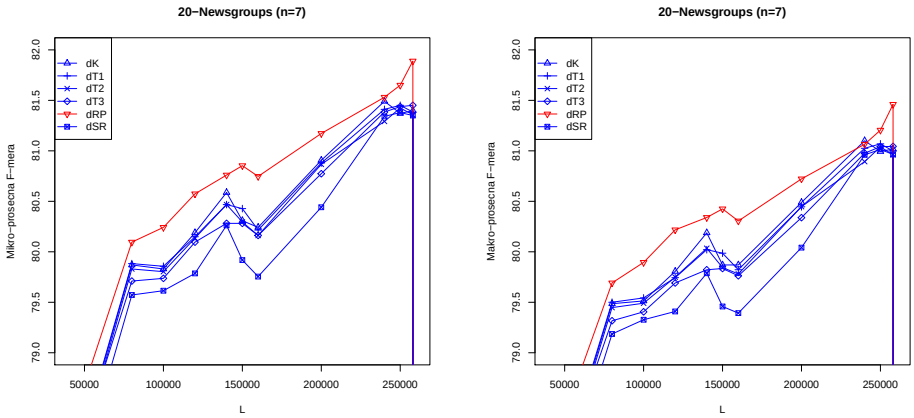


Slika 5.27: Mikro- i makro-prosečna F-mera za 20-Newsgroups korpus, za različite vrednosti parametra n i meru različitosti dSR

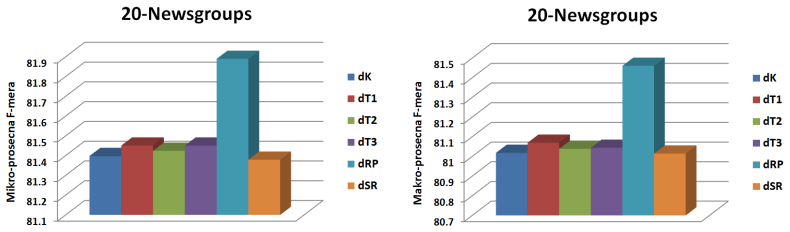
Rezultati su pokazali da se najveća tačnost metode, u slučaju svih mera različitosti, postiže za $n = 7$. Upravo za tu vrednost parametra n izvršeno je poređenje svih mera različitosti u terminima mikro- i makro-prosečne F-mere a rezultati su prikazani na slici 5.28. Maksimalne vrednosti tačnosti za svaku meru različitosti prikazani su na slici 5.29 , a u tabeli 5.18 su prikazne vrednosti parametara n i L za koje se ti maksimumi postižu. Na osnovu rezultata se može zaključiti da na ovom korpusu najveću tačnost postiže mera dRP. Dok se u slučaju Reuters korpusa semantika dokumenta može uhvatiti za relativno male dužine profila L, u slučaju 20-Newsgroup korpusa tačnost metode raste sa porastom dužine profila i dostiže svoj maksimum za veoma velike dužine profila, za sve mere različitosti.

Modifikacija na nivou mere različitosti i profila klase U slučaju modifikacije metode na niovu profila klasa, eksperimentalni rezulati su pokazali da se za $n = 7$ naveća tačnost postiže kada se prilikom generisanja profila klase u razmatranje uzimaju samo oni n-grami koji se pojavljuju u maksimalno 3 od 20 klasa, uključujući i pripadajuću klasu. Na slikama 5.30 i 5.31 je prikazano poređenje osnovne varijante metode i njenih modifikacija za mere različitosti dK i dRP, redom. Za sve ostale mere, poređenje je prikazano na slikama 9 i 10 u Poglavlju 1.

Zaključak koji se može izvesti jeste da se kao i u slučaju Reuters korpusa, tačnost metode značajno povećava kada se izvrši njena modifikacija na nivou profila klase, a vrednost parametra L za koju se postižu optimalni rezultati se



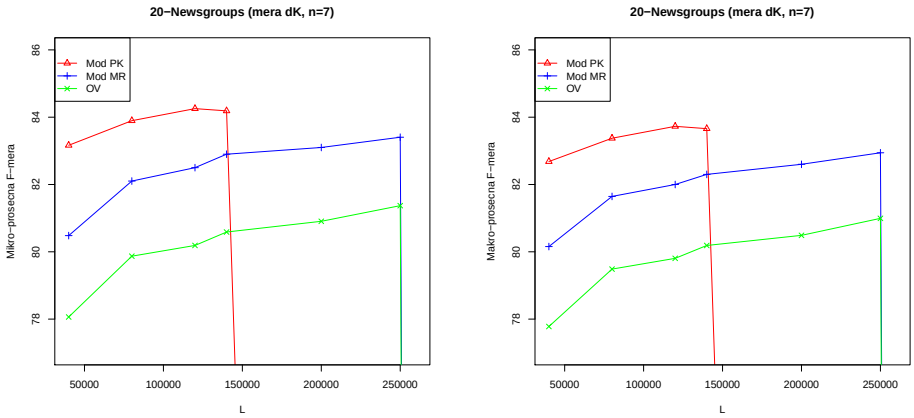
Slika 5.28: Poređenje mera različitosti na 20-Newsgroups korpusu, u terminima mikro- i makro-prosečne F-mere, za $n = 7$.



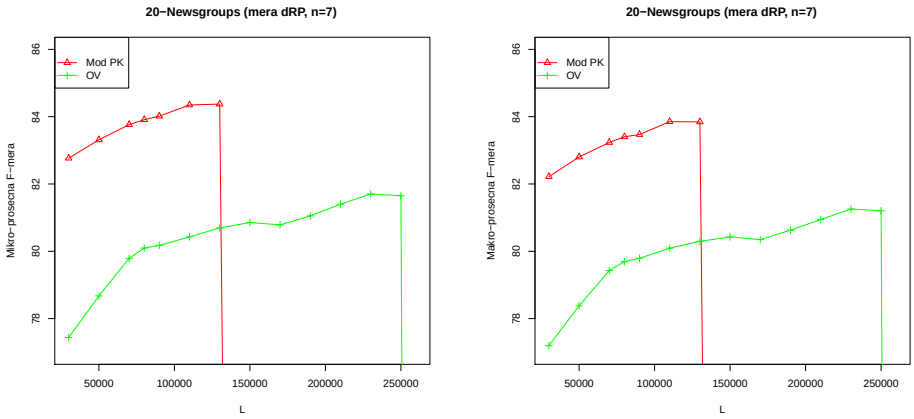
Slika 5.29: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru bez obzira na vrednosti parametara n i L, za korpus 20-Newsgroups i sve mere različitosti.

OV	dK	$dT1$	$dT2$	$dT3$	dRP	dSR
Mikro F-mera	81.3978	81.45097	81.4244	81.45097	**81.8894**	81.37922
n	7	7	7	7	**7**	7
L	258000	250000	250000	258000	**258000**	250000
Makro F-mera	81.0181	81.06937	81.03888	81.04403	**81.4606**	81.01455
n	7	7	7	7	**7**	7
L	258000	250000	250000	258000	**258000**	250000

Tabela 5.18: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za osnovnu varijantu metode zasnovane na n-gramima bajtova, primenjene na Reuters-21578 korpusu.



Slika 5.30: Poređenje osnovne varijante metode sa njenim modifikacijama na 20-Newsgroups korpusu, za $n = 7$ i meru različitosti dK, u terminima mikro- i makro-prosečne F-mere.



Slika 5.31: Poređenje osnovne varijante metode sa njenim modifikacijama na 20-Newsgroups korpusu, za $n = 7$ i meru različitosti dRP, u terminima mikro- i makro-prosečne F-mere.

smanjuje. Tačnost se u terminu mikro-prosečne F-mere povećava sa 81.8894 na 84.37417 a u terminu makro-prosečne F-mere povećava se sa 81.4606 na 83.84507, dok se vrednosti parametra L u oba slučaja smanjuje sa 258000 na 130000. Osim toga, prostor pretrage optimalnog rešenja po parametru L se smanjuje (sa 258001 na 143400) s obzirom na to da se smanjuje maksimalna moguća dužina profila najmanje klase (u ovom korpusu to je "comp.os.ms-

windows.misc") jer se iz profila klase isključuju svi n-grami koji se pojavljuju u više od 3 klase.

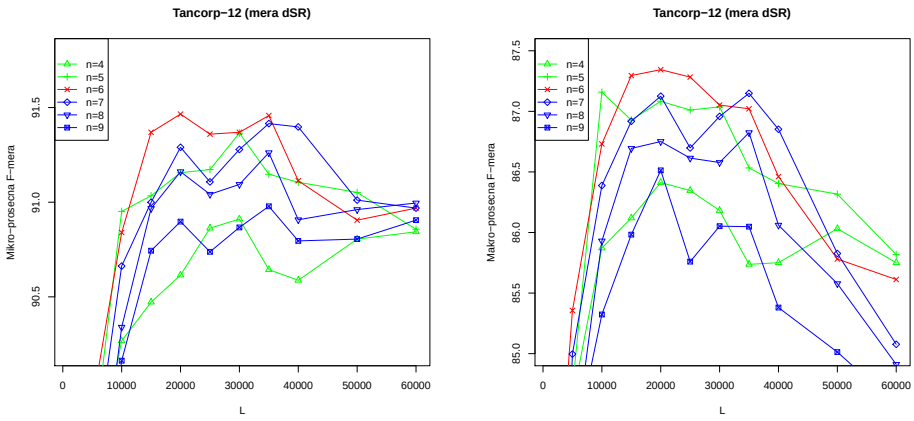
slika max za OV, Mod MR i Mod PK

tabela

Tancorp-12

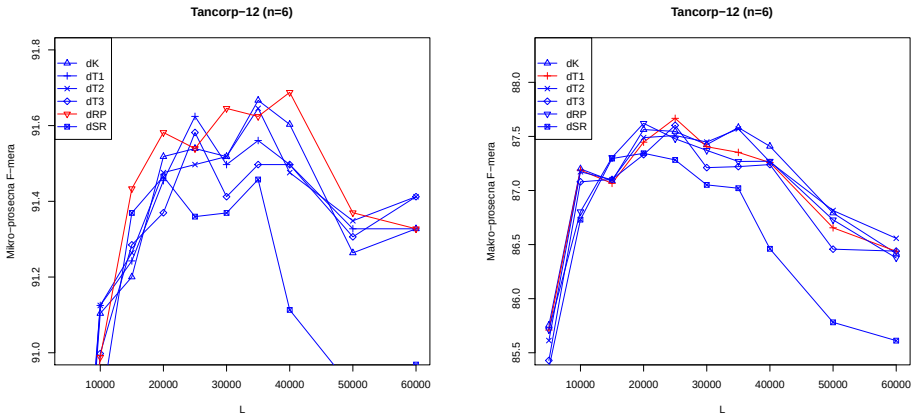
Osim na korpusima na engleskom i srpskom jeziku, veliki izazov predstavlja testiranje metode na korpusu na kineskom jeziku. Kineski jezik je veoma izazovan i po mnogo čemu različit od evropskih jezika. Ujedno, to je maternji jezik najvećem broju ljudi na svetu.

Na slici 5.32 su prikazani rezultati testiranja osnovne varijante metode na Tancorp-12 korpusu za meru dSR, za vrednosti parametara n od 4 do 9 i L od 5000 do 60000 sa korakom 5000. Za ostale mere različitosti, rezultati su prikazani na slikama 15 i 16 u okviru poglavlja Prilog 1.

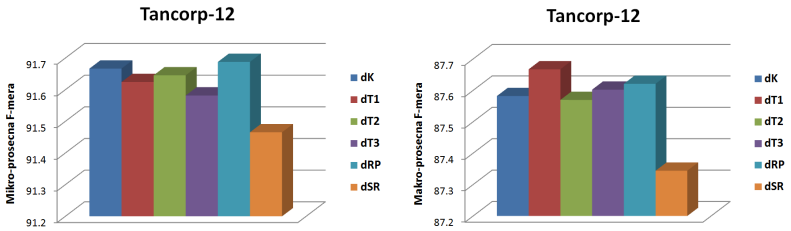


Slika 5.32: Mikro- i makro-prosečna F-mera za osnovnu varijantu metode zasnovane na n-gramima, testiranje na Tancorp-12 korpusu, za različite vrednsti parametra n i meru različitosti dSR.

Sa ovih slika može da se zaključi da se najveća tačnost metode postiže za $n = 6$, za sve mere različitosti. Za tu vrednost parametra n na slici 5.33 je prikazano poređenje svih mera različitosti u terminima mikro- i makro-prosečne F-mere za L od 5000 do 60000 sa korakom 5000 a na slici 5.34 je prikazno poređenje mera različitosti u terminima maksimalnih vrednosti za mikro- i makro-prosečnu F-meru bez obzira na vrednosti parametara n i L. Tabela 5.19 prikazuje koje su to vrednosti ovih parametara za koje se postižu optimalni rezultati.



Slika 5.33: Poređenje mera različitosti na Tancorp-12 korpusu za $n = 6$ i promenljivu vrednost parametra L, u terminima mikro- i makro-prosečne F-mere.



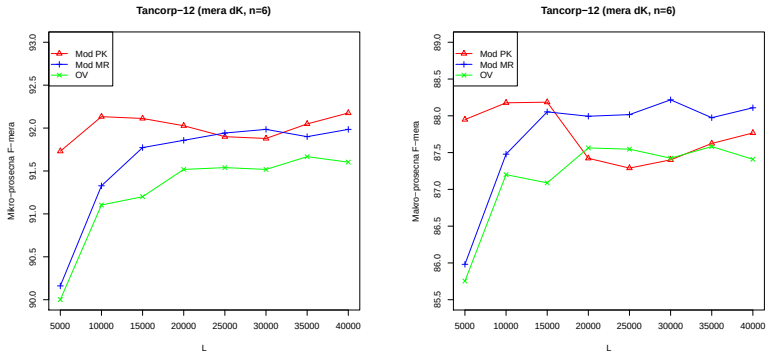
Slika 5.34: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru za korpus Tancorp-12 i sve mere različitosti, bez obzira na vrednosti parametara n i L.

Modifikacija na nivou mere različitosti i profila klase Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu za mere različitosti dK i dRP, prikazano je na slikama 5.35 i 5.36, redom. Za ostale mere različitosti, rezultati su prikazani na slikama 17 i 18 u okviru poglavlja Prilog 1. Prilikom konstruisanja profila klase u okviru modifikacije metode na nivou profila klase, pokazalo se da se najbolji rezultati dobijaju kada se u razmatranje uzimaju samo oni n-grami koji su sadržani u profilima još najviše 3 od 12 klasa, osim pripadajuće.

Poređenje maksimalnih vrednosti za mikro- i makro-prosečnu F-meru (bez obzira na vrednosti parametara n i L) za osnovnu varijantu metode i njene modifikacije, za svaku meru različitosti pojedinačno, prikazano je na slici 5.37. Dodatno, tabele 5.20 i 5.21 prikazuju vrednosti parametara n i L za

OV	*dK*	*dT*1	*dT*2	*dT*3	*dRP*	*dSR*
Mikro F-mera	91.6667	91.6243	91.6455	91.5818	**91.6878**	91.4655
n	6	6	6	6	**6**	6
L	35000	25000	35000	25000	**40000**	20000
Makro F-mera	87.582	**87.6671**	87.5696	87.6023	87.6209	87.3441
n	6	**6**	6	6	6	6
L	35000	**25000**	35000	25000	20000	20000

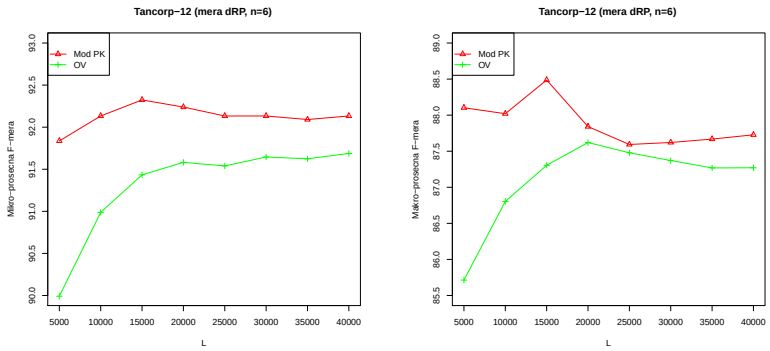
Tabela 5.19: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *osnovnu varijantu metode*, primenjenu na Tancorp-12 korpusu.



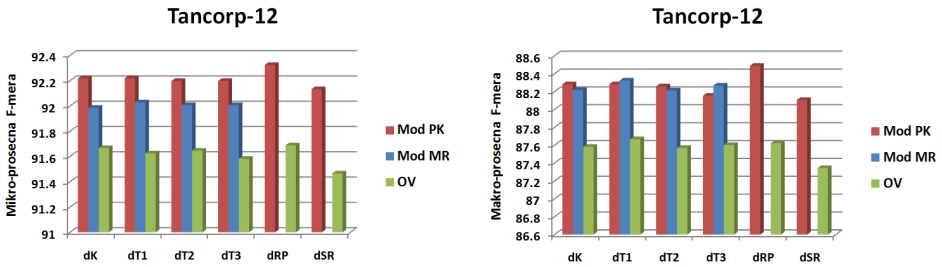
Slika 5.35: Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu, za $n = 6$ i meru različitosti dK, u terminima mikro- i makro-prosečne F-mere.

koje se dobijaju te maksimalne vrednosti.

Može se zaključiti da i u slučaju korpusa na kinsekom jeziku modifikacija na nivou profila klase postiže bolje rezultate (mikro-F mera se povećava sa 91.6878 na 92.324, a makro-F mera sa 87.6671 na 88.4873) uz smanjenju vrednost parametra L za koju se maksimumi tačnosti postižu (za mikro-F meru L se smanjuje sa 40000 na 15000, a za makro-F meru sa 25000 na 15000) kao i značajno smanjenu oblast pretrage optimalnog rešenja po parametru L (maksimalna moguća vrednost za parametar L smanjena je sa 143044 na 76600).



Slika 5.36: Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu, za $n = 6$ i meru različitosti dRP, u terminima mikro- i makro-prosečne F-mere.



Slika 5.37: Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu, za $n = 6$, u terminima maksimalnih vrednosti za mikro- i makro-prosečne F-mere.

Mod MR	dK	$dT1$	$dT2$	$dT3$
Mikro F-mera	91.62426	92.02714	92.00594	92.00594
n	6	6	6	6
L	25000	25000	25000	25000
Makro F-mera	87.66714	88.32324	88.21198	88.26769
n	6	6	6	6
L	25000	25000	25000	25000

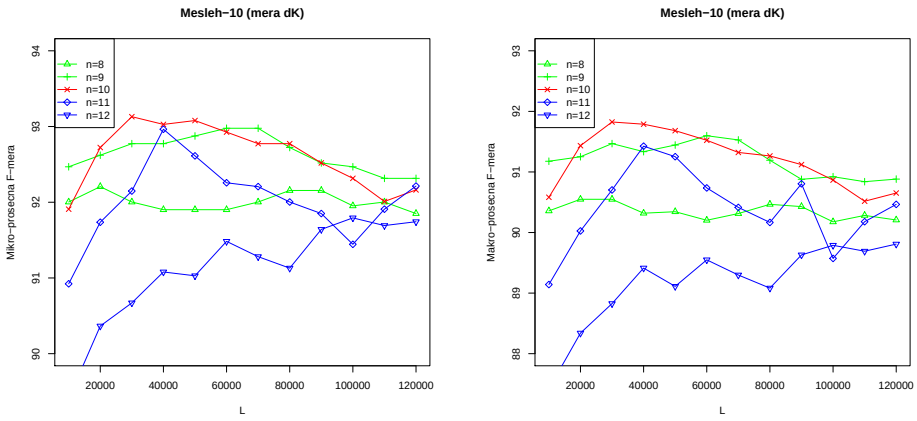
Tabela 5.20: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou mere različitosti*, primenjenu na Tancorp-12 korpusu.

Mod PK	*dK*	*dT*1	*dT*2	*dT*3	*dRP*	*dSR*
Mikro F-mera	92.21798	92.1967	92.1967	92.1967	**92.324**	92.1319
n	6	6	6	6	**6**	6
L	15000	15000	15000	15000	**15000**	15000
Makro F-mera	88.28	88.2579	88.2579	88.1528	**88.4873**	88.1055
n	6	6	6	6	**6**	6
L	15000	15000	15000	15000	**15000**	15000

Tabela 5.21: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou profila klase*, primenjenu na Tancorp-12 korpusu.

Mesleh-10

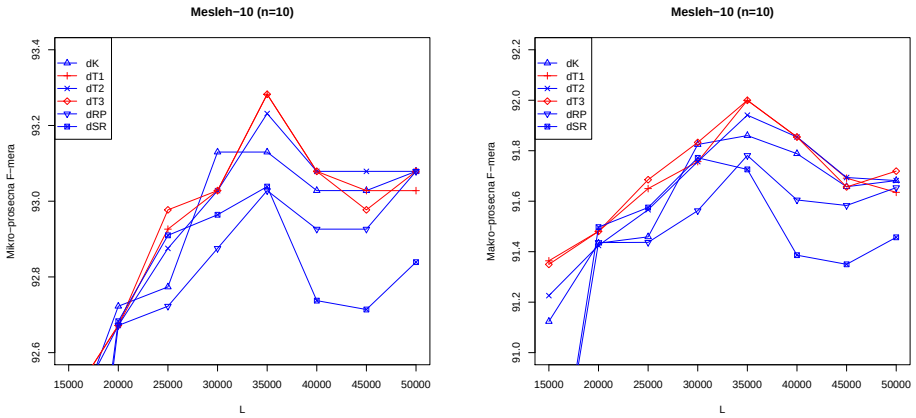
Osim na kineskom, najveći izazov predstavlja testiranje metode na korpusu na arapskom jeziku. U cilju dobijanja vrednosti parametra n za koju se postižu najbolji rezultati, osnovna varijanta metode je testirana za vrednosti parametra n od 8 do 12 i za vrednosti parametra L od 10000 do 120000 sa korakom 10000. Rezultati dobijeni za meru dK prikazani su na slici 5.38. Za sve ostale mere različitosti rezultati su prikazani na slikama 11 i 12 u poglavlju Prilog 1.



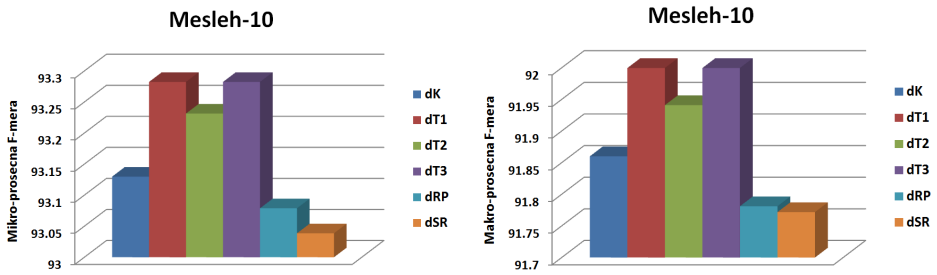
Slika 5.38: Mikro- i makro-prosečna F-mera za Mesleh-10 korpus, za različite vrednosti parametra n i meru različitosti dK

Može se primetiti da se najveća tačnost metode postiže za $n = 10$. Za tu vrednost parametra n izvršeno je poređenje svih ranije pomenutih mera

različitosti a rezultati su prikazani na slici 5.39. Poređenje maksimalnih vrednosti tačnosti za sve mere različitosti bez obzira na vrednosti parametara n i L prikazano je na slici 5.40, a u tabeli 5.22 su dodatno prikazane vrednosti tih parametara za koje se ovi maksimumi postižu.



Slika 5.39: Poređenje mera različitosti na Mesleh-10 korpusu za $n = 6$ i promenljivu vrednost parametra L, u terminima mikro- i makro-prosečne F-mere.



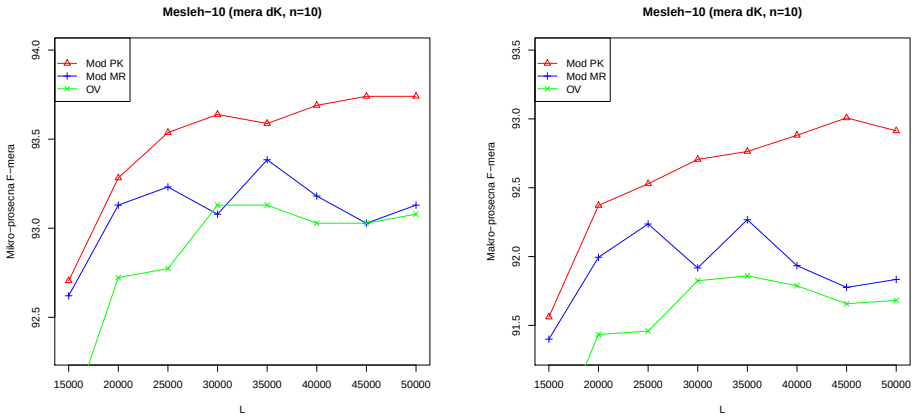
Slika 5.40: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru za korpus Mesleh-10 i sve mere različitosti, bez obzira na vrednosti parametara n i L.

Modifikacija na nivou mere različitosti i profila klase Prilikom konstruisanja profila klase u okviru modifikacije metode na nivou profila klase, pokazalo se da se najbolji rezultati postižu kada se u razmatranje uzimaju samo oni n-grami koji su osim u pripadajućoj klasi, sadržani u još eventualno

OV	*dK*	*dT*1	*dT*2	*dT*3	*dRP*	*dSR*
Mikro F-mera	93.1298	**93.2824**	93.2315	**93.2824**	93.0789	93.0386
n	10	**10**	10	**10**	10	10
L	30000	**35000**	35000	**35000**	50000	35000
Makro F-mera	91.86	**91.9996**	91.9409	**91.9997**	91.7811	91.7718
n	10	**10**	10	**10**	10	10
L	35000	**35000**	35000	**35000**	35000	30000

Tabela 5.22: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *osnovnu varijantu metode*, primenjenu na Mesleh-10 korpusu.

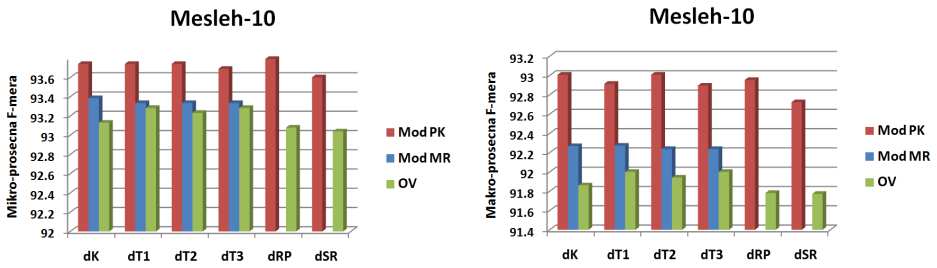
jednoj klasi. Na slici 5.41 je prikazano poređenje osnovne varijante metode sa njenim modifikacijama za meru dK. Za sve ostale mere različitosti, rezultati su prikazani na slikama 13 i 14 u poglavlju Prilog 1.



Slika 5.41: Poređenje osnovne varijante metode sa njenim modifikacijama na Mesleh-10 korpusu, za $n = 10$ i meru različitosti dK, u terminima mikro- i makro-prosečne F-mere.

Poređenje maksimalnih tačnosti dobijenih za svaku meru različitosti primenom osnovne varijante metode i njenih modifikacija, prikazano je na slici 5.42. Tabele 5.23 i 5.24 prikazuju vrednosti parametara n i L za koje se ti maksimumi postižu.

Na osnovu prikazanih rezultata za Mesleh-10 korpus može da se zaključi da modifikacija na nivou profila klase daje bolje rezultate u odnosu na osnovnu varijantu metode. Mikro-F mera se povećava sa 93.2824 na 93.7914 a



Slika 5.42: Poređenje aksimalnih vrednosti za mikro- i makro-prosečnu F-meru osnovne varijante metode i njenih modifikacija, za korpus Mesleh-10 i sve mere različitosti bez obzira na vrednosti parametara n i L.

Mod MR	$dKmod$	$dT1mod$	$dT2mod$	$dT3mod$
Mikro F-mera	**93.3842**	93.3333	93.3333	93.3333
n	**10**	10	10	10
L	**35000**	35000	35000	35000
Makro F-mera	92.2677	**92.2716**	92.2366	92.2366
n	10	**10**	10	10
L	35000	**25000**	25000	25000

Tabela 5.23: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou mere različitosti*, primenjenu na Mesleh-10 korpusu.

Mod PK	dK	$dT1$	$dT2$	$dT3$	dRP	dSR
Mikro F-mera	93.7404	93.7404	93.7404	93.6895	**93.7914**	93.6008
n	10	10	10	10	**10**	10
L	45000	50000	45000	45000	**50000**	50000
Makro F-mera	**93.0076**	92.9136	**93.0076**	92.8948	92.9529	92.7226
n	**10**	10	**10**	10	10	10
L	**45000**	50000	**45000**	45000	50000	50000

Tabela 5.24: Maksimalne vrednosti za mikro- i makro-prosečnu F-meru i vrednosti parametara n i L za koje se ti maksimumi postižu, za *modifikaciju metode na nivou profila klase*, primenjenu na Mesleh-10 korpusu.

makro sa 91.9997 na 93.0076. Za razliku od svih ostalih korpusa, vrednost parametra L za koju se postižu optimalni rezultati se povećava sa 35000 na 50000 za mikro-, odnosno sa 35000 na 45000 za makro-prosečnu F-meru, iako

se prostor pretrage optimalnog rešenja po parametru L smanjuje (maksimalna moguća vrednost parametra L se smanjuje sa 311177 na 281671).

Iz svih do sada prikazanih rezultata, može da se zaključi da se novouvedenom modifikacijom metode na nivou mere različitosti povećava tačnost metode za sve prikazane korpuse sa izuzetkom Tancorp-12 korpusa, na račun jednog izračunavanja težinskih faktora za sve n-grame koji čine profil klase. Više od toga, modifikacijom metode na nivou profila klase, za sve prikazane korpuse bez izuzetka se povećava tačnost metode a optimalni rezultati se dobijaju za manje vrednosti parametra L (osim u slučaju arapskog korpusa), čime se povećava efikasnost metode. Dodatno, prostor pretrage optimalnog rešenja po parametru L se smanjuje za sve korpuse s obzirom na činjenicu da se smanjuje maksimalna moguća dužina profila svih pa i najmanje klase u korpusu.

5.2 Metoda zasnovana na wordnet-u

Primenom procedure klasifikacije opisane u odeljku 4.2, svakoj klasi iz Ebart-3 korpusa pridružuje se lista koncepata, odnosno literala iz srpskog wordnet-a, koji će predstavljati tu klasu. Na osnovu najfrekventnijih reči (imenica i glagola) za svaku klasu, izdvajaju se koncepti iz srpskog wordnet-a koji su kandidati da budu članovi liste predstavnika te klase. Za svaki koncept kandidat se izračunava težina na način koji je opisan u odeljku 4.2. Ukoliko je njegova težina veća od nekog unapred zadatog broja, taj koncept se dodaje u listu predstavnika klase. Pokazalo se da je broj 3 dobar izbor granice za težinu koju koncept kandidat treba da pređe kako bi bio pridružen listi. Za svaki koncept izračunava se i broj aktivnih literala koji su mu dodeljeni, odnosno broj literala sinonima (koji se pojavljuju u bar jednom dokumentu korpusa) pridruženih tom konceptu ili konceptima koji sa njim grade neke od sledećih veza: podređen-nadređen, celina-deo, celina-član, antonimija, relacija koja povezuje značenje i domen korišćenja koncepta i relacija izvođenja. Na ovaj način, svakoj klasi pridruženi su sledeći koncepti:

- EKONOMIJA:
  - "*banka*" ($grupa \rightarrow drusxtvena\ grupa \rightarrow organizacija \rightarrow ustanova \rightarrow finansijska\ institucija \rightarrow banka$)
    - Težina=127.77, broj aktivnih literala je 1.
  - "*kredit*" ($svojina \rightarrow imovina \rightarrow kredit$)
    - Težina=42.48, broj aktivnih literala je 1.

  - "*trzxisxte*" (*uzdrzxati se* → *raditi* → *cyin* → *aktivnost* → *trzxisxte*)
    - Težina=41.74, broj aktivnih literala je 2.
  - "*prodaja*" (*dogadxaj* → *grupna akcija* → *obavlxanxe posla* → *trgovina* → *marketing* → *prodaja*)
    - Težina=19.65, broj aktivnih literala je 5.
  - "*industrija*" (*dogadxaj* → *grupna akcija* → *obavlxanxe posla* → *trgovina* → *poslovanxe* → *industrija*) filtrirano po domenu "enterprise".
    - Težina=19.44, broj aktivnih literala je 5.
  - "*ustanova*" (*grupa* → *drusxtvena grupa* → *organizacija* → *ustanova*) filtrirano po domenima "economy(...)" i "enterprise".
    - Težina=16.80, broj aktivnih literala je 4.
  - "*dug*" (*svojina* → *novcyane obaveze* → *dug*)
    - Težina=10.12, broj aktivnih literala je 2.
  - "*preduzecxe*" (*grupa* → *drusxtvena grupa* → *organizacija* → *preduzecxe* → *preduzecxe*) filtrirano po domenu "enterprise".
    - Težina=4.99, broj aktivnih literala je 26.
  - "*novcyana jedinica*" (*apstrakcija* → *velicyina* → *odredxena velicyina* → *jedinica* → *novcyana jedinica*) filtrirano po domenu "economy".
    - Težina=3.53, broj aktivnih literala je 25.

- POLITIKA:
  - "*izbor*" (*uzdrzxati se* → *raditi* → *cyin* → *akcija* → *izbor* → *izbor*)
    - Težina=63.78, broj aktivnih literala je 1.
  - "*stranka*" (*grupa* → *drusxtvena grupa* → *organizacija* → *stranka*)
    - Težina=30.97, broj aktivnih literala je 3.
  - "*parlament*" (*grupa* → *drusxtvena grupa* → *skupina* → *zakonodavna skupsxtina* → *parlament*)
    - Težina=28.97, broj aktivnih literala je 3.
  - "*medxunarodan*" (*medxunarodan*)

    - Težina=28.94, broj aktivnih literala je 1.
  - "*politika*" (*psihicykosvojstvo* → *saznanxe* → *misaoniproces* → *visxisaznajniproces* → *razmisxlxanxe* → *logicykomisxlxenxe* → *zaklxucyivanxe* → *politika*)
    - Težina=22.69, broj aktivnih literala je 1.
  - "*ministar*" (*entitet* → *predmet* → *zxiva stvar* → *bicxe* → *lxudsko bicxe* → *vodxa* → *odgovorno lice* → *administrator* → *rukovodilac* → *ministar*)
    - Težina=18.52, broj aktivnih literala je 41.
  - "*poglavar*" (*entitet* → *predmet* → *zxiva stvar* → *bicxe* → *lxudsko bicxe* → *saopsxtavacy* → *pregovaracy* → *predstavnik* → *poglavar drzxave*)
    - Težina=12.37, broj aktivnih literala je 5.
  - "*rat*" (*stanxe* → *antagonizam* → *rat*)
    - Težina=10.66, broj aktivnih literala je 1.
  - "*zajednica*" (*grupa* → *drusxtvena grupa* → *zajednica*) filtrirano po domenu "anthropology".
    - Težina=8.90, broj aktivnih literala je 8.
  - "*politicyko telo*" (*grupa* → *drusxtvena grupa* → *organizacija* → *politicyko telo*) filtrirano po domenu "politics".
    - Težina=8.20, broj aktivnih literala je 11.
  - "*koalicija*" (*grupa* → *drusxtvena grupa* → *organizacija* → *koalicija*) filtrirano po domenu "politics".
    - Težina=7.12, broj aktivnih literala je 6.
  - "*glas*" (*uzdrzxati se* → *raditi* → *cyin* → *akcija* → *izbor* → *glas*)
    - Težina=6.47, broj aktivnih literala je 6.
  - "*narod*" (*grupa* → *grupa lxudi* → *narod*) filtrirano po domenu "politics".
    - Težina=4.54, broj aktivnih literala je 6.
  - "*podrsxka*" (*uzdrzxati se* → *raditi* → *cyin* → *aktivnost* → *pomocx* → *podrsxka*) filtrirano po domenu "politics".
    - Težina=4.25, broj aktivnih literala je 15.
- SPORT:

- "*klub*" (*entitet* → *predmet* → *celina* → *lxudska tvorevina* → *konstrukcija* → *zgrada* → *klub*)
  - Težina=30.23, broj aktivnih literala je 2.
- "*sezona*" (*apstrakcija* → *velicyina* → *osnovna velicyina* → *period* → *sezona*)
  - Težina=25.75, broj aktivnih literala je 2.
- "*pobeda*" (*dogadxaj* → *desxavanxe* → *konac* → *pobeda*)
  - Težina=22.44, broj aktivnih literala je 3.
- "*tim*" (*grupa* → *drusxtvena grupa* → *organizacija* → *skupina* → *tim*)
  - Težina=19.25, broj aktivnih literala je 3.
- "*skor*" (*apstrakcija* → *velicyina* → *odredxena velicyina* → *brojna velicyina* → *skor*)
  - Težina=16.86, broj aktivnih literala je 2.
- "*lopta*" (*entitet* → *predmet* → *celina* → *lxudska tvorevina* → *igracyka* → *lopta*)
  - Težina=16.63, broj aktivnih literala je 1.
- "*takmicyenxe*" (*dogadxaj* → *drusxtveni dogadxaj* → *takmicyenxe*)
  - Težina=8.68, broj aktivnih literala je 12.
- "*igra*" (*apstrakcija* → *velicyina* → *igra*) filtrirano po domenu "sport(...)".
  - Težina=7.27, broj aktivnih literala je 13.
- "*oprema*" (*entitet* → *predmet* → *celina* → *lxudska tvorevina* → *instrumetarijum* → *oprema*) filtrirano po domenu "sport(...)".
  - Težina=7.27, broj aktivnih literala je 13.
- "*takmicyar*" (*entitet* → *predmet* → *zxiva stvar* → *bicxe* → *lxudsko bicxe* → *takmicyar*)
  - Težina=5.05, broj aktivnih literala je 19.
- "*sport*" (*uzdrzxati se* → *raditi* → *cyin* → *aktivnost* → *rekreacija* → *sport*)
  - Težina=3.18, broj aktivnih literala je 27.

Pri izboru koncepata potrebno je voditi računa i o tome da ne bude velika razlika u broju aktivnih literala koji se pridružuju klasama kako ne bi došlo

do pogrešnog favorizovanja neke od klasa. U opisanom slučaju broj različitih aktivnih literala za klasu *Ekonomija* je 66, *Politika* je 59 i *Sport* je 69.

Za svaki dokument za testiranje se izračunava njegova mera pripadnosti svakoj od klasa. Dokument se pridružuje onoj klasi za koju ima najveću vrednost mere pripadnosti. Problem koji se pri tome može javiti jeste da jedan dokument može imati maksimalnu vrednost mere pripadnosti za više različitih klasa. Korpus se međutim karakteriše nepreklapajućim klasama, odnosno svaki dokument se može pridružiti tačno jednoj klasi. Optimistički pristup ovom problemu bio bi da ukoliko se klasa kojoj dokument zaista pripada nalazi među klasama za koje dokument postiže maksimum mere pripadnosti, smatra se da je dokument ispravno klasifikovan. Realističan pristup je da se smatra da je taj dokument i ispravno klasifikovan (za klasu kojoj stvarno pripada) i neispravno klasifikovan (za sve ostale klase za koje postiže maksimum a zapravo im ne pripada).

Dobijeni rezultati za optimistički pristup prikazani su u tabeli 5.25 a za realističan pristup u tabeli 5.26.

	TP	FP	FN	Preciznost	Odziv	F-mera
Sport	450	37	38	92.40	92.21	92.31
Ekonomija	130	34	36	79.27	78.31	78.79
Politika	425	45	42	90.43	91.01	90.72
Makro-prosek				87.37	87.18	87.27
Mikro-prosek				89.65	89.65	89.65

Tabela 5.25: Rezultati klasifikacije Ebart-3 korpusa metodom zasnovanom na wordnet-u – optimistički pristup.

	TP	FP	FN	Preciznost	Odziv	F-mera
Sport	450	66	38	87.21	92.21	89.64
Ekonomija	130	103	36	55.79	78.31	65.16
Politika	425	85	42	83.33	91.01	87.00
Makro-prosek				75.45	87.18	80.60
Mikro-prosek				78.15	90.00	79.83

Tabela 5.26: Rezultati klasifikacije Ebart-3 korpusa metodom zasnovanom na wordnet-u – realističan pristup.

Poređenja radi, izvršena je klasifikacija korpusa na osnovu domena pridruženih konceptima. Tako su klasama *Ekonomija*, *Politika* i *Sport* pridruženi

svi koncepti, odnosno literali iz srpskog wordnet-a, kojima su u engleskom wordnetu pridružena obeležja domena koji su od interesa za date klase. Ovi domeni su pobrojani u poglavlju 4.2. Na ovaj način dobijaju se rezultati klasifikacije prikazani u tabeli 5.27 (realističan pristup).

	TP	FP	FN	Preciznost	Odziv	F-mera
Sport	376	46	112	89.10	77.05	82.64
Ekonomija	152	302	14	33.48	91.57	49.03
Politika	335	85	132	79.76	71.73	75.54
Makro-prosek				67.45	80.12	69.07
Mikro-prosek				66.59	76.98	66.59

Tabela 5.27: Rezultati klasifikacije Ebart-3 korpusa metodom zasnovanom na domenima u wordnet-u.

Broj aktivnih literala u ovom slučaju za klasu *Ekonomija* je 249, za klasu *Politika* je 111 a za klasu *Sport* je 66. Iz tabele može da se zaključi da je klasa *Ekonomija*, između ostalog i time što joj je dodeljen veći broj aktivnih literala, pogrešno favorizovana (302 dokumenta su joj pogrešno dodeljena). Zbog toga je izvršen još jedan eksperiment gde je broj aktivnih literala po klasi ograničen na 66 (koliko ih ima u klasi *Sport*), ali onih koji pripadaju konceptima sa najvećom frekvencijom u posmatranoj klasi. Dobijeni su rezultati prikazani u tabeli 5.28.

	TP	FP	FN	Preciznost	Odziv	F-mera
Sport	402	51	86	88.74	82.38	85.44
Ekonomija	147	274	19	34.92	88.55	50.09
Politika	365	108	102	77.17	78.16	77.66
Makro-prosek				66.94	83.03	71.06
Mikro-prosek				67.85	81.53	67.85

Tabela 5.28: Rezultati klasifikacije Ebart-3 korpusa metodom zasnovanom na domenima u wordnet-u, sa ograničenim brojem aktivnih literala.

Iz svega ovoga može da se izvede zaključak da se pametnim izborom koncepata metodom opisanom u poglavlju 4.2 dobijaju bolji rezultati nego metodom zasnovanom na domenima iz wordnet-a. U slučaju metode zasnovane na domenima, bolji rezultati se dobijaju ako se broj aktivnih literala pridruženih klasama ograniči u cilju njihovog ujednačavanja za sve klase.

Smatra se da bi rezultati dobijeni metodom zasnovanom na wordnet-u bili znatno bolji ukoliko bi korpus bio takav da ga čine duži dokumenti (da

sadrže veći broj reči) jer bi tada bilo manje slučajeva da je mera pripadnosti nekog dokumenta ista za više od jedne klase (bila bi manja razlika između optimističkog i realističkog pristupa). Takođe, ova metoda bi dala bolje rezulate kada bi se primenila na korpusu koji koristi nestandardan i proširen vokabular.

6. Poređenje sa postojećim metodama

U ovom poglavlju biće prikazano poređenje modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova (označavaćemo je sa "kNN bajt-ngrami") sa drugim do sada objavljenim metodama. Kako bi poređenje bilo što ispravnije i ubedljivije, prilikom klasifikacije metodom prikazanom u ovom radu korišćeni su isti korpusi i izvršena je ista podela na skupove za učenje i testiranje kao kod metoda sa kojima se vrši poređenje. Do sada najpoznatiji način predstavljanja tekstualnog dokumenta je "vreća reči" (eng. "bag of words", BOW), gde se dokument predstavlja skupom reči koje se pojavljuju u bar jednom dokumentu. Metoda "kNN bajt-ngrami" prikazana u ovom doktoratu se zasniva na predstavljanju dokumenta preko n-grama bajtova pa je zanimljivo izvršiti poređenje dobijenih rezultata ovom metodom sa rezultatima drugih metoda zasnovanih na n-gramima kao i sa drugim metodama zasnovanim na BOW reprezentaciji teksta.

6.1 Poređenje sa drugim n-gram metodama

U slučaju 20-Newsgroups i Reuters-21578 korpusa, poređenje je izvršeno sa metodom zasnovanom na n-gramima koja je predstavljena u radu čiji su autori Rahmoun i Elberrichi [62]. U ovom radu autori su koristili n-grame karaktera za reprezentaciju teksta. Prilikom generisanja profila klasa, kako bi selektovali one n-grame koji su od značaja, koristili su χ^2 test. Kao mere različitosti koristili su kosinusnu i Kullback&Liebler meru. U radu je izvršeno poređenje sa reprezentacijom teksta zasnovanom na reči (BOW) i reprezentacijom zasnovanom na korenu reči (eng. stem) gde se udružuju reči sa istim korenom. Autori su pokazali da se reprezentacijom teksta zasnovanom na n-gramima dobijaju bolji rezultati u odnosu na druge dve vrste reprezentacije teksta. Korpusi koji su korišćeni za engleski jezik su 20-Newsgroups i Reuters-21578.

Osim sa rezultatima prikazanim u radu [62], za Reuters-21578 korpus

rezultati se porede i sa rezulatima koje je prikazao Makoto Suzuki sa svojim kolegama u radu [74]. U ovom radu autori su predstavili nov metod zasnovan na "modelu vektorskog prostora". U okviru ove metode dokumenti se ne predstavljaju preko profila već kao vektori *tfidf* vrednosti za n-grame karaktera.

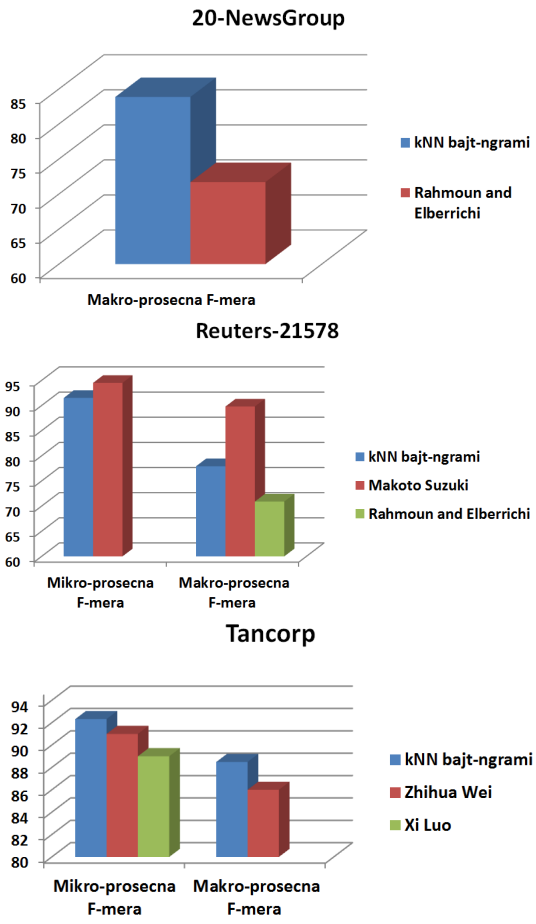
Kad je reč o Tancorp-12 korpusu, poređenje je izvršeno sa rezultatima koje su postigli Zhihua Wei u radu [91] i Xi Luo u radu [49]. Zhihua Wei i kolege u radu [91] koriste kombinacije 1-grama, 2-grama i 3-grama karaktera za reprezentaciju dokumenta kao i njihove apsolutne i relativne frekvencije. Pokazali su da kombinacije 1-grama, 2-grama daju malo bolje rezultate od 1-grama, 2-grama i 3-grama karaktera kao i da relativna frekvencija ne daje bolje rezultate od apsolutne. U radu [49] autori koriste 1-grame i 2-grame karaktera za predstavljanje dokumenata, kao i njihove kombinacije. Primenili su nekoliko pristupa za selekciju i ekstrakciju atributa. Pokazali su da su imenice najznačajnije za predstavljanje dokumenata na kineskom jeziku. Zaključili su i da relativna frekvencija zajedno sa adekvatnom obradom atributa daje poboljšanje rezultata u odnosu na apsolutnu frekvenciju.

Sva ova poređenja prikazana su na slici 6.1. Sa ove slike može da se zaključi da modifikacija metode na nivou profila klase zasnovane na n-gramima bajtova, u slučaju 20-Newsgroups i Tancorp korpusa postiže bolje rezultate od drugih metoda zasnovanih na n-gramima.

6.2 Poređenje sa drugim BOW metodama

Poređenje u slučaju korpusa Reuters-21578 i 20-Newsgroups, izvršeno je sa objavljenim rezultatima autora Man-a [46]. Glavni cilj rada je analiza nekoliko različitih metoda otežavanja atributa, odnosno reči, pri predstavljanju dokumenta i ispitivanje njihovog uticaja na ishod klasifikacije dokumenata, a glavni doprinos je predstavljanje nove metode za pridruživanje težina atributima. Kao metode mašinskog učenja za klasifikaciju teksta korišćene su metoda k-najbližih suseda (kNN) i metoda podržavajućih vektora (SVM). Rezulati poređenja prikazani su na slici 6.2. Sa ove slike može da se zaključi da metoda prikazana u ovom doktoratu postiže bolje rezultate na 20-Newsgroups korpusu u poređenju sa ostalim metodama.

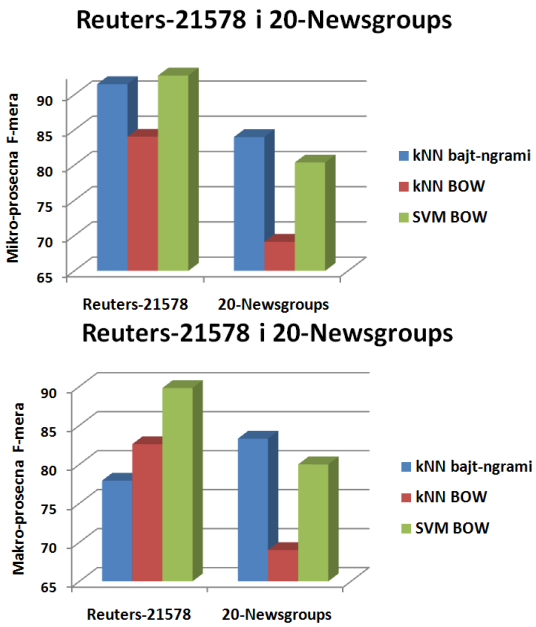
U slučaju Tancorp-12 korpusa poređenje je izvršeno sa Tan-ovim rezultatima objavljenim u radu [76]. U cilju povećanja tačnosti klasifikacije, uvedena je nova strategija nazvana *DragPushing* koja poboljšava kvalitet klasifikatora na kom se primenjuje. U ovom radu, strategija je primenjena na Centroid [28] i Naivnom Bajesovom klasifikatoru. Rezultati klasifikacije na Tancorp-12 korpusu su osim za Naivnu Bajesovu i Centroid metodu prikazani i za metode



Slika 6.1: Poređenje tačnosti modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova ("kNN bajt-ngrami") sa drugim metodama zasnovanim na n-gramima.

k najbližih suseda, metod podržavajućih vektora i Winnow [81] metodu. Na slici 6.3 je prikazano poređenje metode prikazane u ovom doktoratu sa rezultatima pomenutih metoda prikazanih u radu [76]. Može se zaključiti da se jedino metodom podržavajućih vektora postižu bolji rezultati u odnosu na metodu zasnovanu na n-gramima bajtova.

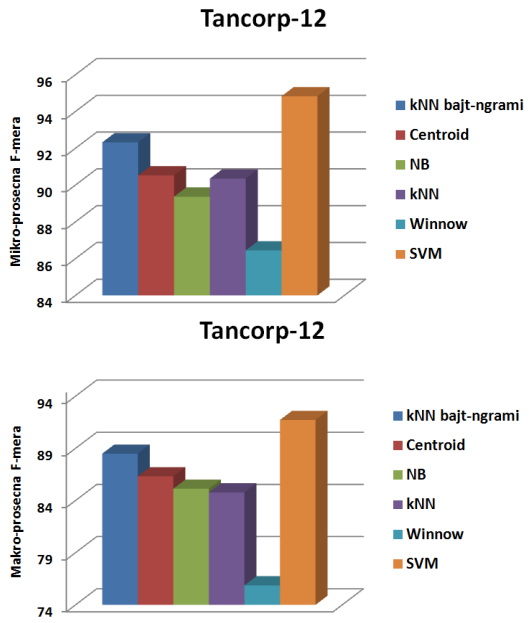
U slučaju Mesleh-10 korpusa, izvršeno je poređenje metode zasnovane na n-gramima bajtova sa rezultatima koje je objavio Mesleh u radu [50]. U ovom radu prikazano je poređenje 17 različitih metoda selekcije atributa. Za sam proces klasifikacije korišćeni su metod podržavajućih vektora, Naivni Bajesov klasifikator, k-najbližih suseda i Rocchio algoritam. Rezultati poređenja su prikazani na slici 6.4. Može se zaključiti da metoda koja je prikazana u ovom



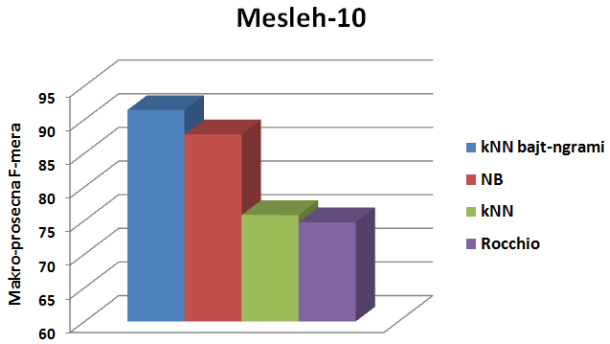
Slika 6.2: Poređenje tačnosti modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova ("kNN bajt-ngrami") sa drugim metodama zasnovanim na BOW reprezentaciji teksta, za Reuters-21578 i 20-Newsgroups korpuse.

doktoratu daje bolje rezultate od svih drugih metoda. Poređenje je izvršeno samo u terminima makro-prosečne F-mere iz razloga što nad ovim korpusom nema objavljenih rezultata za druge metode u terminima mikro-prosečne F-mere.

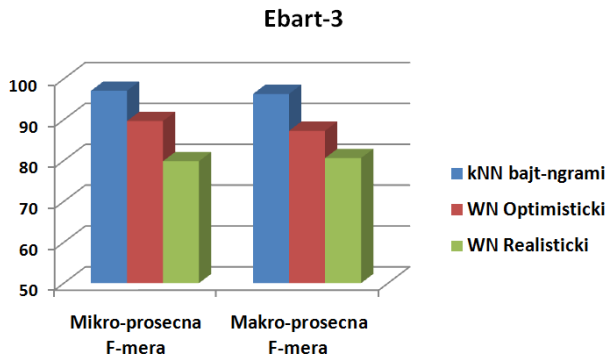
Kad je reč o Ebart-3 korpusu, kako nad ovim korpusom nema objavljenih rezultata drugih metoda, izvršeno je poređenje dve metode prikazane u ovom doktoratu – metode zasnovane na n-gramima bajtova i metode zasnovanom na wordnet-u (optimistički i realistički pristup). Slika 6.5 prikazuje rezultate ovog poređenja i navodi na zaključak da metoda zasnovana na n-gramima bajtova postiže bolju tačnost u odnosu na metodu zasnovanu na wordnet-u.



Slika 6.3: Poređenje tačnosti modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova ("kNN bajt-ngrami") sa drugim metodama zasnovanim na BOW reprezentaciji teksta, za Tancorp-12 korpus.



Slika 6.4: Poređenje tačnosti modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova ("kNN bajt-ngrami") sa drugim metodama zasnovanim na BOW reprezentaciji teksta, za Mesleh-10 korpus.



Slika 6.5: Poređenje tačnosti modifikacije na nivou profila klase metode zasnovane na n-gramima bajtova ("kNN bajt-ngrami") sa sa metodom zasnovanom na wordnet-u ("WN Optimisticki" i "WN Realisticki"), za Ebart-3 korpus.

7. Zaključak

U svetu u kome je znanje moć, sposobnost pretvaranja sirovih podataka u znanje je od neprocenjive vrednosti. Usled stalnog porasta dostupnih količina podataka i potrebe za znanjem skrivenim u njima, oblast *Istraživanje podataka* dobija sve više na značaju. Poslednjih godina ova oblast je doživela veliku ekspanziju zahvaljujući brzom razvoju računarske tehnologije jer je tek razvitkom brzih računarskih sistema postalo moguće efikasno pretraživati ogromne količine sirovih podataka.

Jedan od osnovnih problema koji se rešavaju u okviru *Istraživanja podataka* je klasifikacija koja je i tema ove doktorske disertacije.

U glavi 1 dat je pregled oblasti klasifikacije podataka, a posebno klasifikacije tekstualnih dokumenata prema sadržaju, napisanih prirodnim jezikom. Opisani su različiti načini predstavljanja dokumenata zasnovanih na analizi teksta na različitim nivoima (od nivoa dela reči do pragmatičkog nivoa). Prikazani su različiti korpusi klasifikovanih dokumenata na različitim jezicima: srpskom, engleskom, kineskom i arapskom. Ovi korpusi se razlikuju prema broju i veličinama klasa, prema tome da li su klase preklapajuće ili ne, prema raspodeli dokumenata po klasama, prema veličinama dokumenata i drugo. Glavna razlika je što su napisani na jezicima koji se po svojoj morfologiji i mnogim drugim osobinama međusobno veoma razlikuju. Svi ovi korpusi biće korišćeni za testiranje metode klasifikacije prikazane u ovom radu.

Glavni akcenat u radu stavljen je na klasifikaciju dokumenata na srpskom jeziku koji spada u grupu morfološki složenih jezika. Različiti leksički resursi za srpski jezik (elektronski rečnik, srpski wordnet, korpusi srpskog jezika) i tehnologije njihove obrade prikazani su u okviru glave 2 ovog rada. Ideja je da se ispita kako se informacije sadržane u ovim resursima mogu efikasno iskoristiti u procesu klasifikacije teksta.

U glavi 3 je dat detaljan pregled već postojećih tehnika mašinskog učenja koje se godinama uspešno koriste u procesu klasifikacije podataka. Neke od njih su Drveta odlučivanja, k-Najbližih suseda, Bajesova metoda, Skriveni Markovljevi modeli, Veštačke neuronske mreže i Metoda podržavajućih vektora. Navedeni su i sistemi za klasifikaciju zasnovani na ovim metodama.

Osnovni doprinos ove disertacije izložen je u glavama 4-6. On se sastoji u definisanju novih metoda klasifikacije teksta (glava 4), evaluaciji rezultata koji se postižu primenom ovih metoda (glava 5), kao i poređenju ovih metoda sa drugim najsavremenijim metodama (glava 6). Prva metoda se zasniva na n-gramskoj analizi teksta koja se u osnovi bazira na Keš̆eljevoj metodi za određivanje autorstva teksta [38]. U osnovnoj varijanti metode, n-gramima iz profila klase se pridružuje samo informacija o njihovoj učestalosti u pripadajućoj klasi, ne uzimajući u obzir druge klase. Na nov i originalan način, n-gramima iz profila klasa pridružuju se težinski faktori koji nose informaciju o značaju tog n-grama za tu klasu, uzimajući u obzir i sve druge klase. Ukoliko se n-gram pojavljuje u manjem broju klasa značajniji je za klasu kojoj pripada. Ovo uvođenje težinskih faktora dovelo je do modifikacije metode na dva načina: *modifikacija na nivou mere različitosti* i *modikacija na nivou profila klasa*. Kod *modifikacije na nivou mere različitosti*, profili dokumenata za testiranje ostaju neizmenjeni dok se u profilima klasa svakom n-gramu dodaje kao podatak i novouvedeni težinski faktor. Na osnovu njega se vrši modifikacija mera različitosti tako što se u okviru njihovog izračunavanja frekvencije n-grama iz profila klasa množe novouvedenim težinskim faktorima. Time se veći značaj daje n-gramima koji se pojavljuju u manjem broju klasa. Druga modifikacija je *modifikacija na nivou profila klasa*. Umesto da se posmatra L najfrekventnijih n-grama koji će određivati klasu, posmatra se L najfrekventnijih onih n-grama koji su ekskluzivni za tu klasu, odnosno koji se pojavljuju u toj i eventualno u još nekoj od klasa (optimalan broj klasa u kojima je dopušteno pojavljivanje n-grama zavisi od konkretnog korpusa). To se postiže tako što se iz profila klasa isključuju svi n-grami koji nisu ekskluzivni za tu klasu a onda se posmatra prvih L najfrekventnijih. Ovim se moguća dužina profila klasa smanjuje i time se smanjuje prostor pretrage optimalnog rešenja po parametru L. Štaviše, eksperimentalni rezultati su pokazali da se za sve korpuse optimalni rezultati dobijaju za dosta manje vrednosti parametra L u odnosu na osnovnu varijantu metode i njenu modifikaciju na nivou mere različitosti. To je veoma značajno uzimajući u obzir činjnicu da je glavni nedostatak korišćenja n-gramske metode upravo veliki broj mogućih n-grama. U okviru ove modifikacije koriste se mere različitosti iz osnovne varijante metode. Važno je napomenuti da ova metoda i njene modifikacije mogu biti definisane ne samo za n-grame bajtova već i za n-grame karaktera i reči. Druga metoda je zasnovana na odabranim konceptima iz srpskog wordneta pridruženim klasama. Koncepti se biraju na osnovu vrednosti mere značaja koncepta za klasu. U obzir se uzimaju samo koncepti koji imaju meru veću od nekog unapred definisanog praga. Na osnovu ovih koncepata, za dati dokument se izračunava mera njegove pripadnosti klasama na osnovu čega se dokumentu dodeljuje klasa za koju on ima najveću vrednost

mere pripadnosti. Problem fleksije je rešen korišćenjem srpskog elektronskog rečnika.

U okviru glava 5 i 6 prikazani su rezultati novih metoda i njihovo poređenje sa drugim, do sada objavljenim rezultatima dobijenih primenom različitih metoda nad istim korpusima sa istom podelom na skupove za učenje i testiranje. U slučaju prve metode zasnovane na n-gramima, na srpskom korpusu je izvršeno poređenje ove metode i njenih modifikacija za n-grame na nivou bajta, karaktera i reči. Pokazalo se da prednost imaju n-grami na nivou bajta, ne samo zbog svojih rezultata već i zbog dodatne jezičke nezavisnosti koja se postiže njihovim korišćenjem – nezavisnost od kodne šeme i azbuke. Ovo je i demonstrirano prikazom postignutih rezultata za osnovnu varijantu metode i njenih modifikacija na korpusima na srpskom, engleskom, kineskom i arapskom jeziku. Eksperimenti su pokazali da se dobijaju rezultati uporedivi sa najefikasnijim metodama mašinskog učenja, mada je n-gramska metoda neuporedivo jednostavnja.

Rezultati primene osnovne varijante metode na korpusu na srpskom jeziku prikazani su u radu [21] izloženom na konferenciji. Jezička nezavisnost osnovne varijante metode ilustrovana je na korpusima na srpskom, engleskom i kineskom jeziku. Rezultati su prikazani u radu [23] koji je prihvaćen za publikovanje u časopisu na SCI listi, i radu [22] izloženom na konferenciji. Modifikacija na nivou mere različitosti testirana je na arapskom korpusu a rezultati su prikazani u radu [25]. Modifikacija na nivou profila klase, osim veće efikasnosti, donela je i bolje rezulate u odnosu na osnovnu varijantu metode i njenu modifikaciju na nivou mere različitosti. Rezultati dobijeni primenom ove metode na korpuse na srpskom, engleskom, kineskom i arapskom jeziku prikazani su u radu [24]. Ova metoda, s obzirom da je zasnovana na n-gramima bajtova, ima veoma široku primenu. Na primer, može se primeniti na klasifikaciju različitih tipova dokumenata predstavljenih u digitalnom obliku, ne samo tekstualnih. Dokumenti se mogu klasifikovati na primer, po sadržaju, tipu dokumenta ili autoru. Veoma važna primena ove metode, čiji bi rezultati bili od značaja za širu zajednicu, jeste klasifikacija govora zdravih osoba i obolelih od neurodegenerativnih oboljenja i mogućnost predviđanja stepena oboljenja, a sve to u cilju ranog otkrivanja ili predikcije oboljenja. Prvi koraci su već preduzeti u tom smeru za obolele od Alchajmera a cilj je proširiti istraživanje i na neke druge vrste bolesti koje kao simptome imaju neke promene u govoru. Druga metoda klasifikacije koja je opisana u ovom radu zasnovana je na leksičko-semantičkoj mreži wordnet. Rezultati početne analize wordnet-a objavljeni su u radu [26], gde su definisane mere produktivnosti koncepata koje određuju koliko neki koncept efektivno predstavlja hijerarhiju kojoj pripada. Dodatno je izvršeno istraživanje kako se wordnet može efikasno iskoristiti u cilju bolje klasifikacije tekta. Početni rezultati

klasifikacije teksta korišćenjem srpskog wordneta prikazani su na konferenciji u radu [56] a u radu [27] su prikazani rezultati unapređene metode.

Literatura

[1] Hanne Andersen, Peter Barker, and Xiang Chen. Kuhn's mature philosophy of science and cognitive psychology. *Philosophical Psychology*, 9(3):347–363, 1996.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[3] Michael W Berry. *Survey of text mining I: clustering, classification, and retrieval*, volume 1. Springer, 2004.

[4] David C Blair. Information retrieval and the philosophy of language. *Annual review of information science and technology*, 37(1):3–50, 2003.

[5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[6] Béatrice Bouchou, Mickael Tran, and Denis Maurel. Towards an XML representation of proper names and their relationships. In *Natural Language Processing and Information Systems*, pages 44–55. Springer, 2005.

[7] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[9] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, 2005.

[10] Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine learning*, 10(1):57–78, 1993.

[11] Blandine Courtois, Max Silberztein Ladl, et al. Dictionnaires électroniques du français. *Langue française*, 87(1):3–4, 1990.

[12] T De Heer. Experiments with syntactic traces in information retrieval. *Information Storage and Retrieval*, 10(3):133–144, 1974.

[13] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, 2004.

[14] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[15] Thierry Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *Systems, Man and Cybernetics, IEEE Transactions on*, 25(5):804–813, 1995.

[16] J.S. Downie. *Evaluating a Simple Approach to Musical Information Retrieval: Conceiving Melodic N-grmas as Text*. PhD thesis, University of Western Ontario, 1999.

[17] Matematički fakultet. *Resursi srpskog jezika*. http://korpus.matf.bg.ac.rs.

[18] Christiane Fellbaum. *WordNet*. Springer, 2010.

[19] George Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[20] William J Frawley, Gregory Piatetsky-Shapiro, and Christopher J Matheus. Knowledge discovery in databases: An overview. *AI magazine*, 13(3):57, 1992.

[21] Jelena Graovac. Serbian text categorization using byte level n-grams. In *BCI (Local)*, pages 93–96, 2012.

[22] Jelena Graovac. Text categorization using n-gram based language independent technique. In *35 godina računarske lingvistike u Srbiji, Book of Abstracts, to appear in the Proceedings of the Conference*, 2014.

[23] Jelena Graovac. A variant of n-gram based language-independent text categorization. *Intelligent Data Analysis*, 18(4), 2014.

[24] Jelena Graovac and Gordana Pavlović-Lažetić. Improvements of n-gram based language independent text categorization technique. Rad je u fazi pripreme.

[25] Jelena Graovac and Gordana Pavlović-Lažetić. Text categorization using byte-level n-grams with weighting factors: Facing the challenge of Arabic. Rad je predat časopisu na razmatranje.

[26] Jelena Graovac and Gordana Pavlović-Lažetić. Productivity of concepts in Serbian Wordnet. In *Proceedings of the Sixth Language Technologies Conference*, pages 86–91, 2008.

[27] Jelena Graovac and Gordana Pavlović-Lažetić. Wordnet-based document classification. *Journal of Information and Library Science - INFOtheca*, 2014.

[28] Eui-Hong Sam Han and George Karypis. *Centroid-based document classification: Analysis and experimental results*. Springer, 2000.

[29] Phillip J Hayes, Peggy M Andersen, Irene B Nirenburg, and Linda M Schmandt. Tcs: a shell for content-based text categorization. In *Artificial Intelligence Applications, 1990., Sixth Conference on*, pages 320–326. IEEE, 1990.

[30] Birger Hjørland and Karsten Nissen Pedersen. A substantive theory of classification for information retrieval. *Journal of documentation*, 61(5):582–597, 2005.

[31] Thorsten Joachims. *Learning to classify text using support vector machines: methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

[32] Karen Sparck Jones. Some thoughts on classification for retrieval. *Journal of Documentation*, 26(2):89–101, 1970.

[33] Karen Sparck Jones. What is the role of nlp in text retrieval? In *Natural language information retrieval*, pages 1–24. Springer, 1999.

[34] Karen Spärck Jones. Revisiting classification for retrieval. *Journal of documentation*, 61(5):598–601, 2005.

[35] Jussi Karlgren. *Stylistic experiments in information retrieval*. Springer, 1999.

[36] George Karypis and Eui-Hong Sam Han. Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 12–19. ACM, 2000.

[37] Gordon V Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.

[38] Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264, 2003.

[39] Cvetana Krstev. Specifični koncepti Balkana u semanticčkoj mreži Wordnet. In *Zbornik radova Susreti kultura*, 2004.

[40] Cvetana Krstev. *Processing of Serbian: automata, texts and electronic dictionaries*. Faculty of Philology of the University, 2008.

[41] Cvetana Krstev, Bojana Đorđević, Sanja Antonić, et al. Kooperativan rad na dogradnji srpskog wordneta. *INFOteka: časopis za informatiku i bibliotekarstvo*, 9(1):57–75, 2008.

[42] Cvetana Krstev, Gordana Pavlović-Lažetić, Ivan Obradović, and Duško Vitas. Corpora issues in validation of Serbian Wordnet. In *Text, Speech and Dialogue*, pages 132–137. Springer, 2003.

[43] Cvetana Krstev, Gordana Pavlovic-Lazetic, Duško Vitas, and Ivan Obradovic. Using textual and lexical resources in developing Serbian Wordnet. *Romanian Journal of Information Science and Technology*, 7(1-2):147–161, 2004.

[44] Cvetana Krstev, Ranka Stanković, Duško Vitas, and Ivan Obradović. WS4LR: A workstation for lexical resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC*, pages 1692–1697, 2006.

[45] Cvetana Krstev, Ranka Stankovic, Dusko Vitas, and Ivan Obradovic. The usage of various lexical resources and tools to improve the performance of web search engines. In *LREC*, 2008.

[46] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):721–735, 2009.

[47] Ken Lang. Newsweeder: Learning to filter netnews. In *In Proceedings of the Twelfth International Conference on Machine Learning*. Citeseer, 1995.

[48] David D Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 246–254. ACM, 1995.

[49] Xi Luo, Wataru Ohyama, Tetsushi Wakabayashi, and Fumitaka Kimura. A study on automatic chinese text classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 920–924. IEEE, 2011.

[50] Abdelwadood Mesleh. Feature sub-set selection metrics for Arabic text classification. *Pattern Recognition Letters*, 32(14):1922–1929, 2011.

[51] George A Miller. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[52] Tom Mitchell. *Machine Learning*. McGraw Hill, 1996.

[53] Sreerama K Murthy, Simon Kasif, Steven Salzberg, and Richard Beigel. Oc1: A randomized algorithm for building oblique decision trees. In *Proceedings of AAAI*, volume 93, pages 322–327. Citeseer, 1993.

[54] Ivan Obradović et al. Application of Intex in refinement and validation of Serbian Wordnet. In *6th Intex Workshop*.

[55] G Pavlović-Lažetić. Electronic resources of Serbian: Serbian Wordnet. In *36th International Slavic Conference, MSC, Belgrade, Serbia*, 2006.

[56] Gordana Pavlović-Lažetić and Jelena Graovac. Ontology-driven conceptual document classification. In *International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, pages 383–386, 2010.

[57] István Pilászy. Text categorization and support vector machines. In *the Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*. Citeseer, 2005.

[58] Odile Piton, Denis Maurel, and Claude Belleil. The prolex data base: Toponyms and gentiles for nlp. Technical report, Université Panthéon-Sorbonne (Paris 1), 1998.

[59] Ljubomir Popović and Duško Vitas. Konspekt za izgradnju referentnog korpusa standardnog srpskog jezika. *Naučni sastanak slavista u Vukove dane.*, pages 221–227, 2003.

[60] Martin F Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[61] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[62] Abdellatif Rahmoun and Zakaria Elberrichi. Experimenting n-grams in text categorization. *Int. Arab J. Inf. Technol.*, 4(4):377–385, 2007.

[63] Manuel de Buenaga Rodríguez, Jose Maria Gomez Hidalgo, and Belen Diaz Agudo. Using Wordnet to complement training information in text categorization. *arXiv preprint cmp-lg/9709007*, 1997.

[64] Fabrice Rossi and Nathalie Villa. Classification in Hilbert spaces with support vector machines. *Proceedings of ASMDA*, pages 635–642, 2005.

[65] Paolo Rosso, Edgardo Ferretti, Daniel Jiménez, and Vicente Vidal. Text categorization and information retrieval using Wordnet senses. In *The Second Global Wordnet Conference GWC*, 2004.

[66] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[67] Sam Scott and Stan Matwin. Text classification using Wordnet hypernyms. In *Proceedings in Use of WordNet in Natural Language Processing Systems*, pages 38–44, 1998.

[68] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[69] Victor V Solovyev and Kira S Makarova. A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localization. *Computer applications in the biosciences: CABIOS*, 9(1):17–24, 1993.

[70] Ranka Stanković and Ivan Obradović. Integracija heterogenih tekstualnih resursa. In *Zbornik radova medunarodnog simpozijuma Razlike izmedu bosanskog/bosnjackog, hrvatskog i srpskog jezika.*, pages 596–616, 2007.

[71] Ranka Stankovic, Ivan Obradovic, Cvetana Krstev, and Duško Vitas. Production of morphological dictionaries of multi-word units using a multipurpose tool.

[72] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

[73] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[74] Makoto Suzuki, Naohide Yamagishi, Yi-Ching Tsai, Takashi Ishida, and Masayuki Goto. English and Taiwanese text categorization using n-gram based on vector space model. In *Information Theory and its Applications (ISITA), 2010 International Symposium on*, pages 106–111. IEEE, 2010.

[75] Pang-Ning Tan et al. *Introduction to data mining*. Pearson Education India, 2007.

[76] Songbo Tan, Xueqi Cheng, Moustafa M Ghanem, Bin Wang, and Hongbo Xu. A novel refinement approach for text categorization. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 469–476. ACM, 2005.

[77] Andrija Tomović and Predrag Janičić. A variant of n-gram based language classification. In *AI* IA 2007: Artificial Intelligence and Human-Oriented Computing*, pages 410–421. Springer, 2007.

[78] Andrija Tomović, Predrag Janičić, and Vlado Kešelj. N-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer methods and programs in biomedicine*, 81(2):137–153, 2006.

[79] Dan Tufis, Dan Cristea, and Sofia Stamou. BalkaNet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information science and technology*, 7(1-2):9–43, 2004.

[80] Miloš Utvić, Ranka Stanković, and Ivan Obradović. Integrisano okruženje za pripremu paralelizovanog korpusa. *Die Unterschiede zwischen dem Bosnischen/Bosniakischen, Kroatischen und Serbischen*, pages 563–578, 2008.

[81] P.P.T.M van Mun. Text classification in information retrieval using Winnow, 1999.

[82] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.

[83] Duško Vitas. Lokalne gramatike srpskog jezika. *Zbornik Matice srpske za slavistiku.*, pages 305–317, 2007.

[84] Duško Vitas, Svetla Koeva, Cvetana Krstev, and Ivan Obradović. Tour du monde through the dictionaries. In *Actes du 27eme Colloque International sur le Lexique et la Gammaire*, pages 249–256, 2008.

[85] Duško Vitas and Cvetana Krstev. Srpski jezik i SNTPI. Technical report, Faculty of Mathematics, University of Belgrade.

[86] Dusko Vitas and Cvetana Krstev. Literature and aligned texts. *Readings in Multilinguality, Bulgarian Academy of Sciences, Sofia, Bulgaria.*, pages 148–155, 2007.

[87] Duško Vitas, Cvetana Krstev, and Denis Maurel. A note on the semantic and morphological properties of proper names in the prolex project. *Lingvisticae Investigationes*, 30(1):115–133, 2007.

[88] Duško Vitas, Gordana Pavlović-Lažetić, Cvetana Krstev, Ljubomir Popović, and Ivan Obradović. Processing Serbian written texts: an overview of resources and basic tools. In *Workshop on Balkan Language Resources and Tools*, volume 21, pages 97–104, 2003.

[89] Piek Vossen. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Boston, 1998.

[90] PJTM Vossen. Introduction to the special issue on the BalkaNet project. 2004.

[91] Zhihua Wei, Duoqian Miao, Jean-Hugues Chauchat, and Caiming Zhong. Feature selection on Chinese text classification using character n-grams. In *Rough Sets and Knowledge Technology*, pages 500–507. Springer, 2008.

[92] Janusz L Wiśniewski. Effective text compression with simultaneous digram and trigram encoding. *Journal of Information Science*, 13.

[93] Derick Wood. Standard generalized markup language: Mathematical and philosophical issues. In *Computer Science Today*, pages 344–365. Springer, 1995.

[94] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

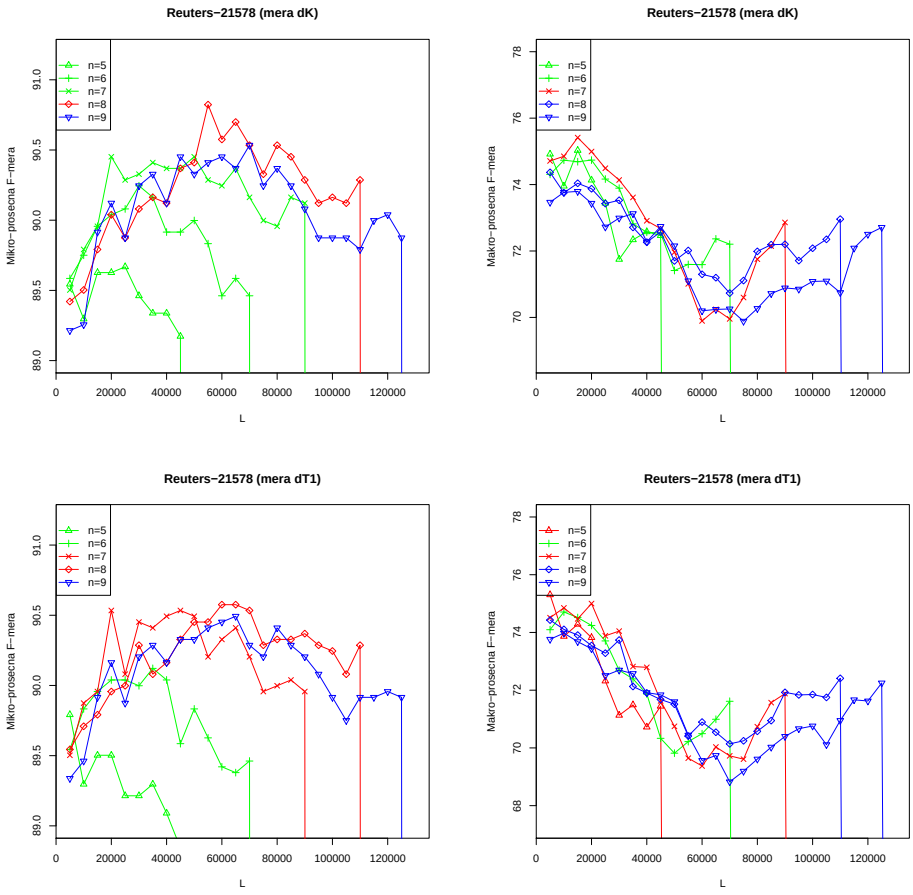
[95] Institut za standardizaciju Srbije. *Terminološki rad, Vokabular, Deo 2: Primene računara*. Oznaka standarda: SRPS ISO 1087-2:2005, Beograd, 2005.

[96] EM Zamora, Joseph J Pollock, and Antonio Zamora. The use of trigram analysis for spelling error detection. *Information Processing & Management*, 17(6):305–316, 1981.

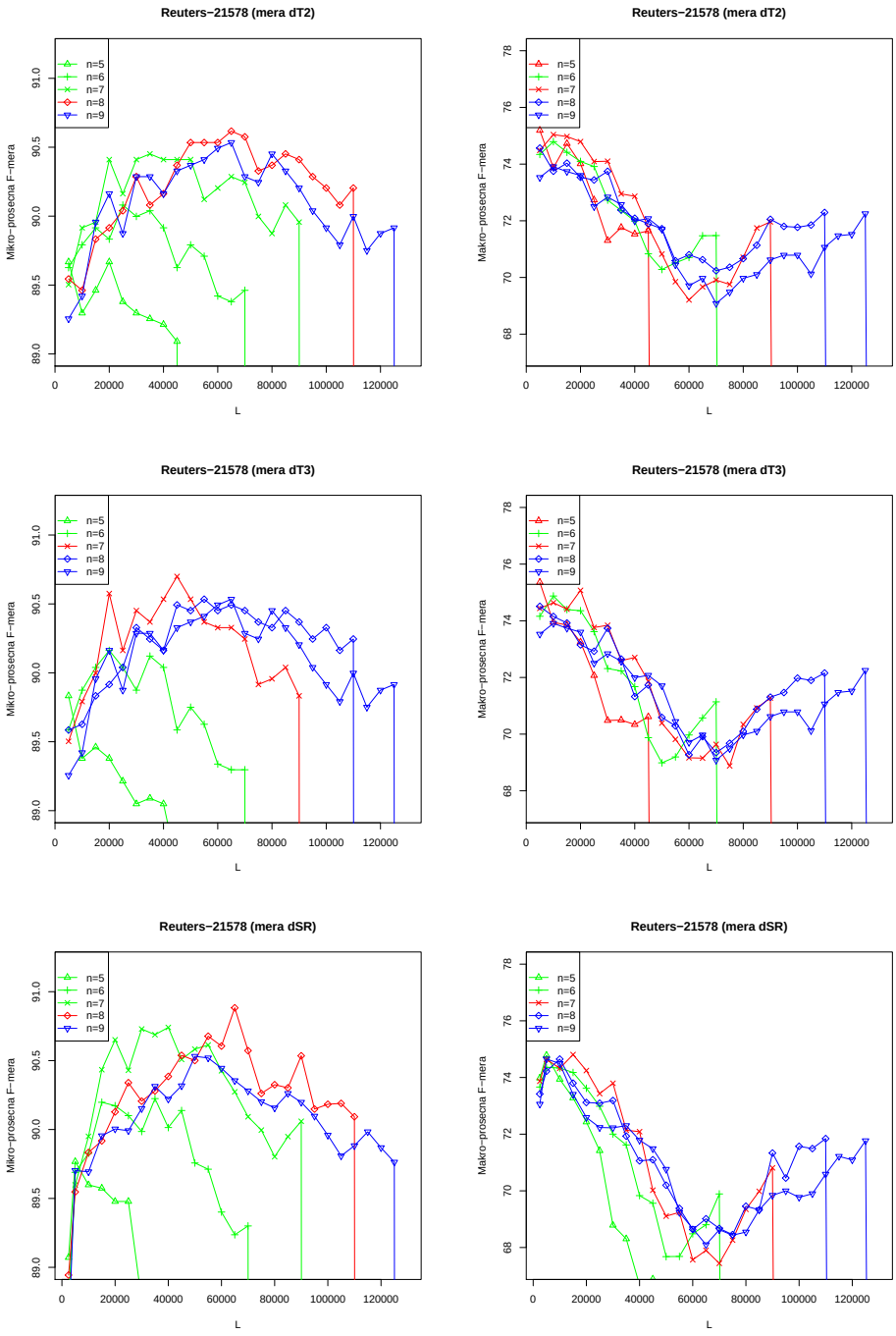
[97] Joseph Zernik. Data mining as a civic duty - online public prisoners registration systems. *SOCIAL MEDIA*, page 84, 2010.

Prilog 1

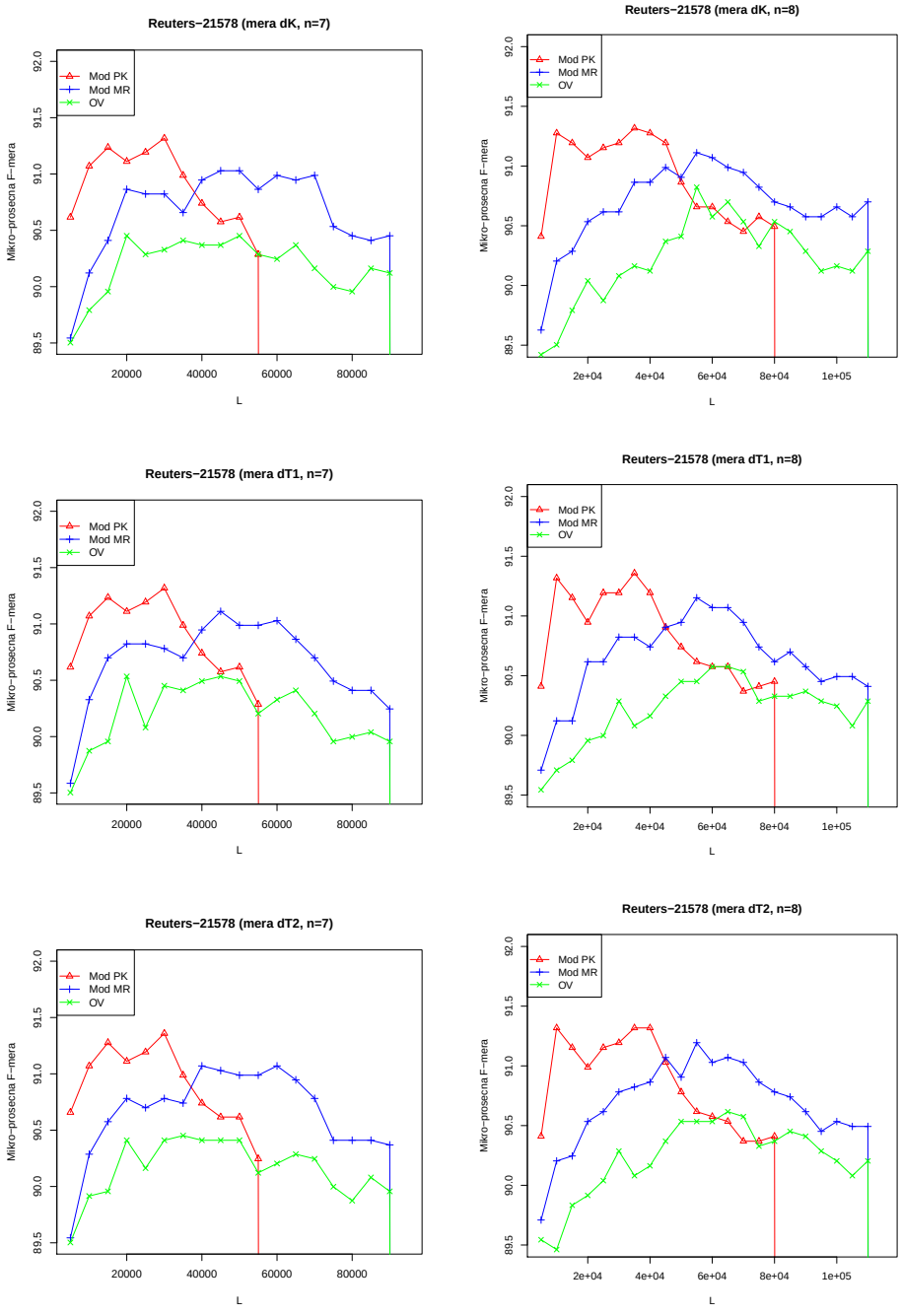
Reuters-21578



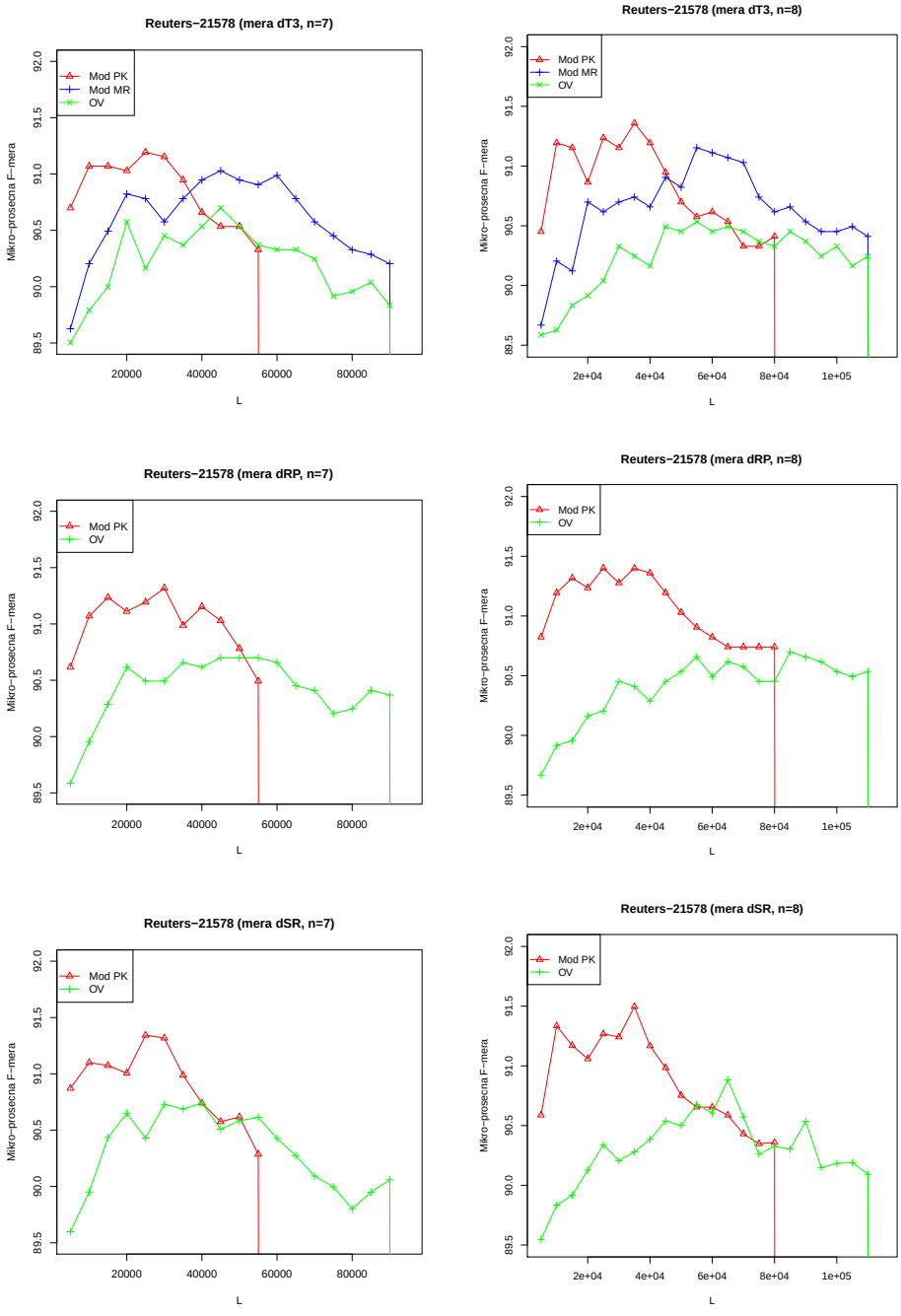
Slika 1: Mikro- i makro-prosečna F-mera za Reuters-21578 korpus, za različite vrednosti parametra n i mere različitosti dK i $dT1$.



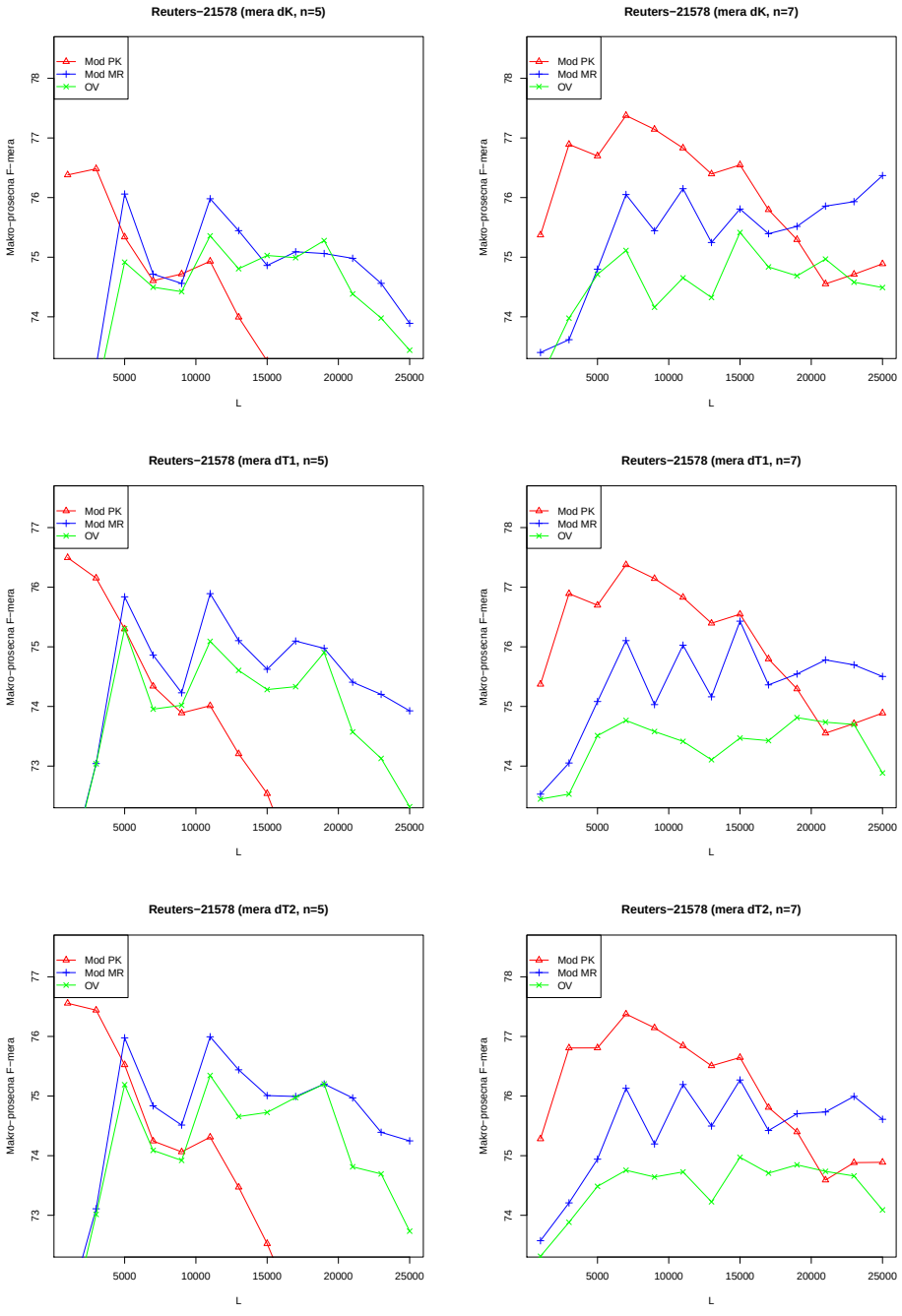
Slika 2: Mikro- i makro-prosečna F-mera za Reuters-21578 korpus, za različite vrednosti parametra n i mere različitosti $dT2$, $dT3$ i dSR.



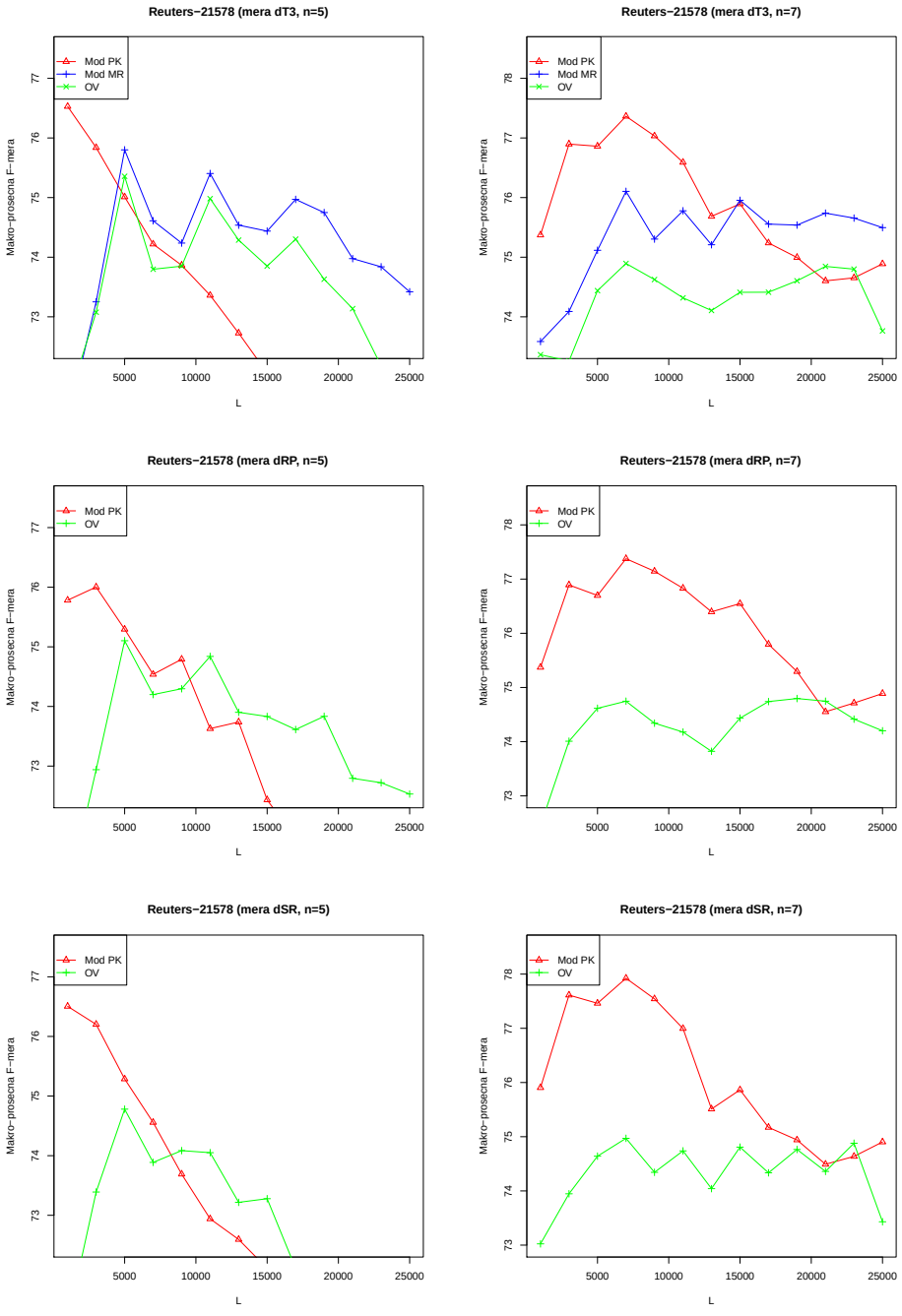
Slika 3: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za $n = 7$ i $n = 8$ i mere različitosti dK, $dT1$ i $dT2$, u terminu mikro-prosečne F-mere.



Slika 4: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za $n = 7$ i $n = 8$ i mere različitosti $dT3$, dRP i dSR, u terminu mikro-prosečne F-mere.

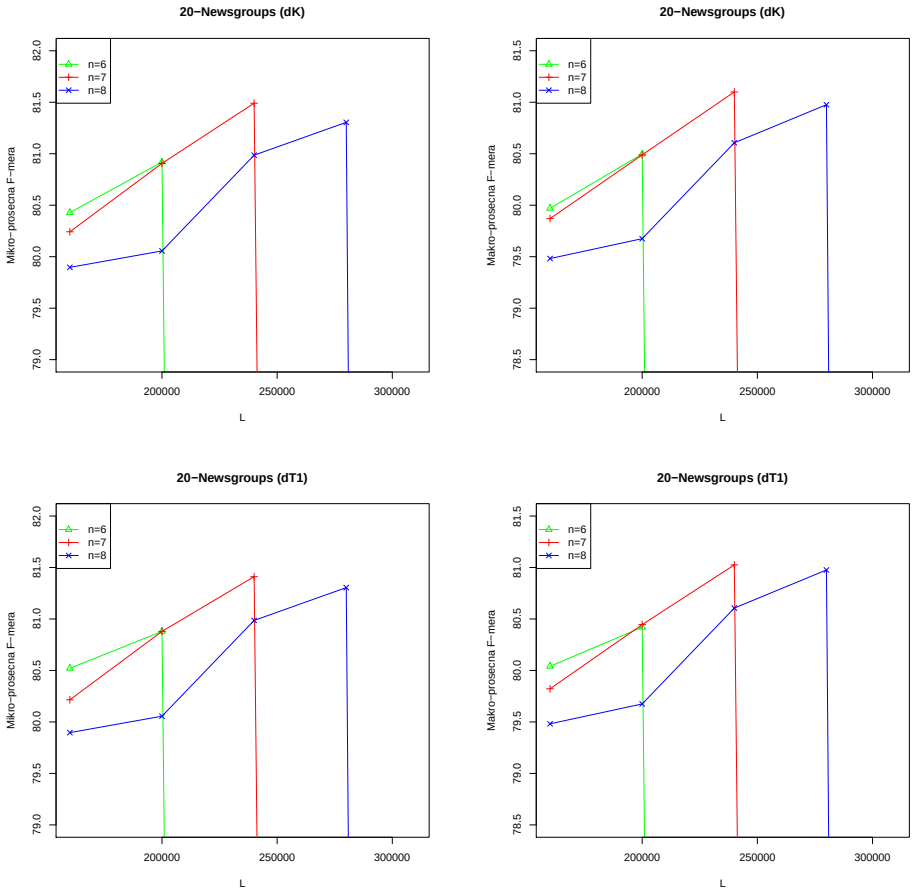


Slika 5: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za $n = 5$ i $n = 7$ i mere različitosti dK, $dT1$ i $dT2$, u terminu makro-prosečne F-mere.

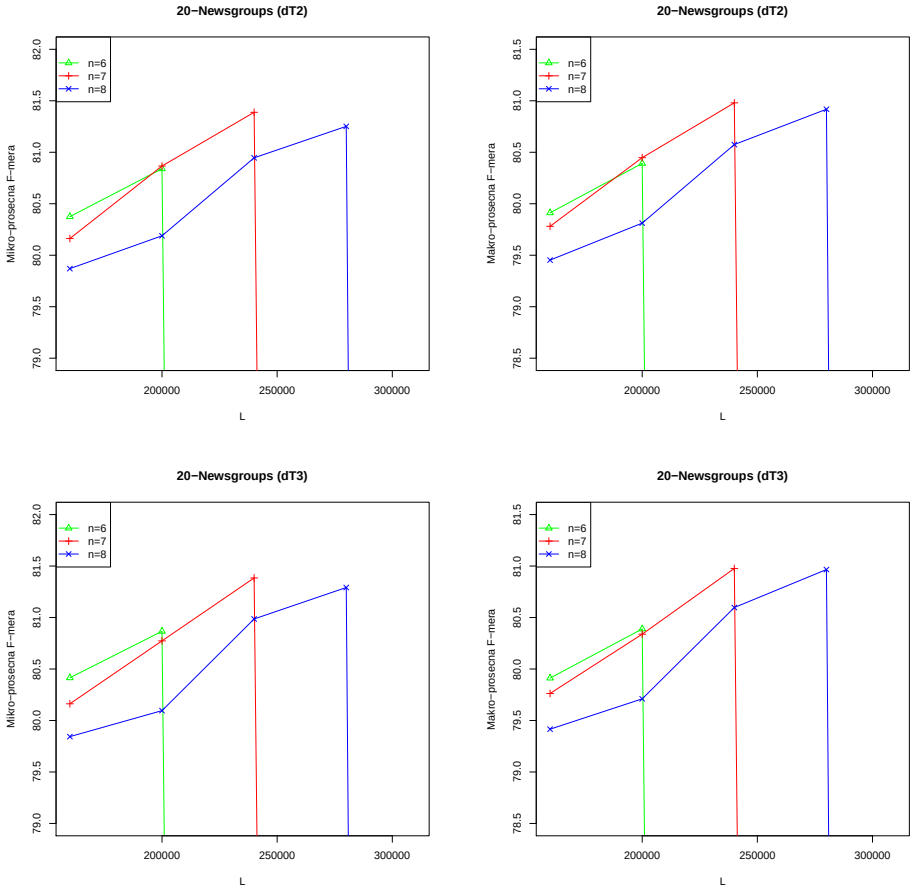


Slika 6: Poređenje osnovne varijante metode sa njenim modifikacijama na Reuters-21578 korpusu, za $n = 5$ i $n = 7$ i mere različitosti $dT3$, dRP i dSR, u terminu makro-prosečne F-mere.

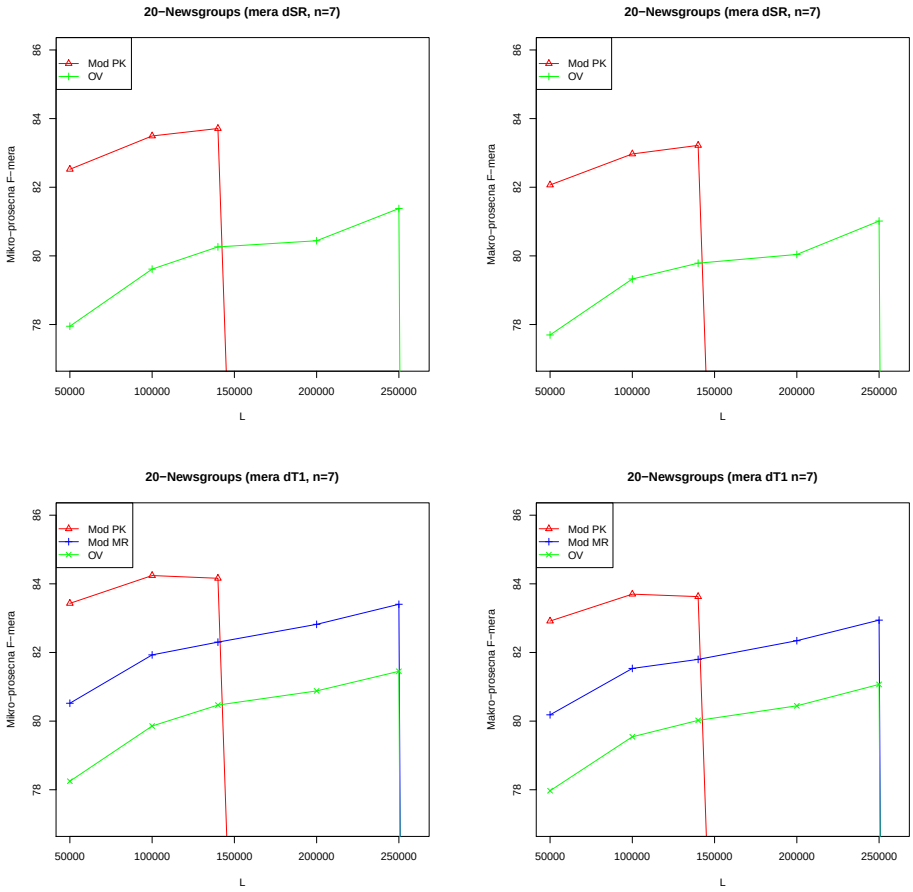
20-Newsgroups



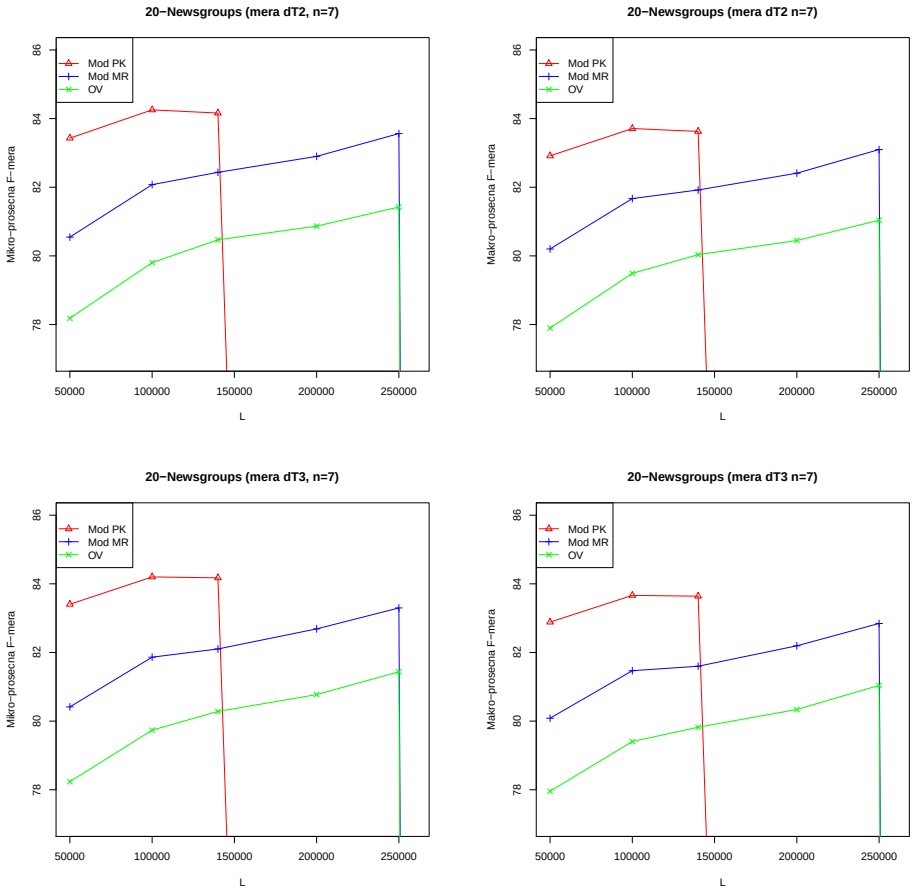
Slika 7: Mikro- i makro-prosečna F-mera za 20-Newsgroups korpus, za različite vrednosti parametra n i mere različitosti dK i $dT1$.



Slika 8: Mikro- i makro-prosečna F-mera za 20-Newsgroups korpus, za različite vrednosti parametra n i mere različitosti $dT2$ i $dT3$.

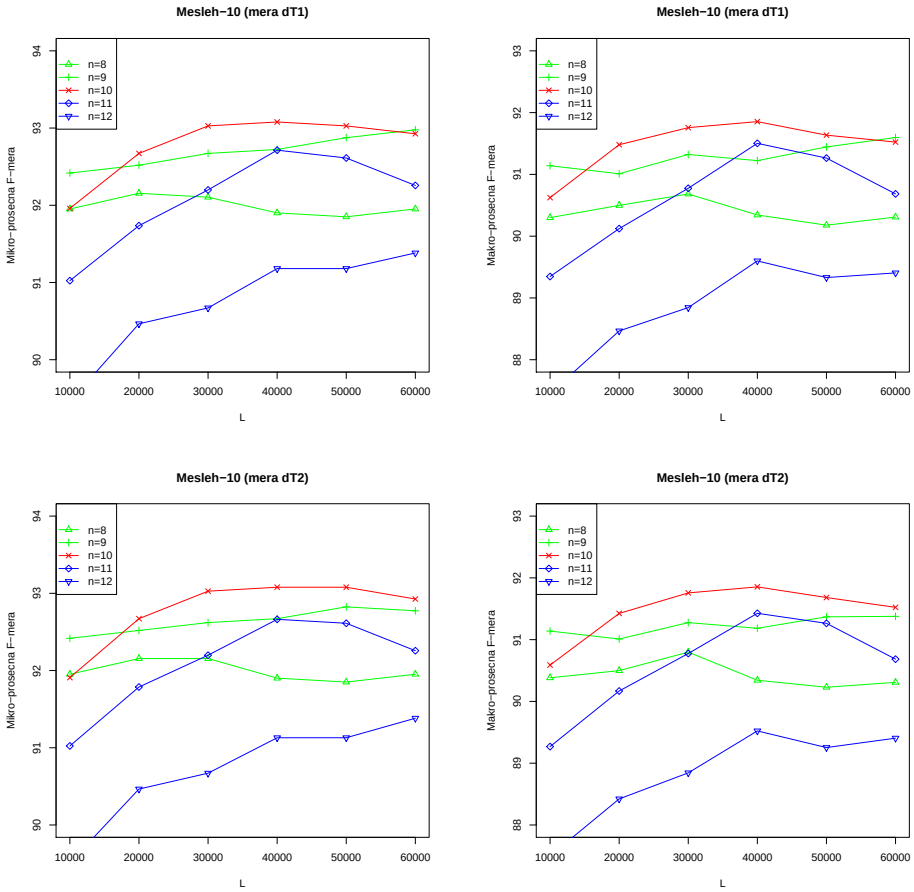


Slika 9: Poređenje osnovne varijante metode sa njenim modifikacijama na 20-Newsgroups korpusu, za $n = 7$ i mere različitosti dSR i $dT1$, u terminima mikro- i makro-prosečne F-mere.

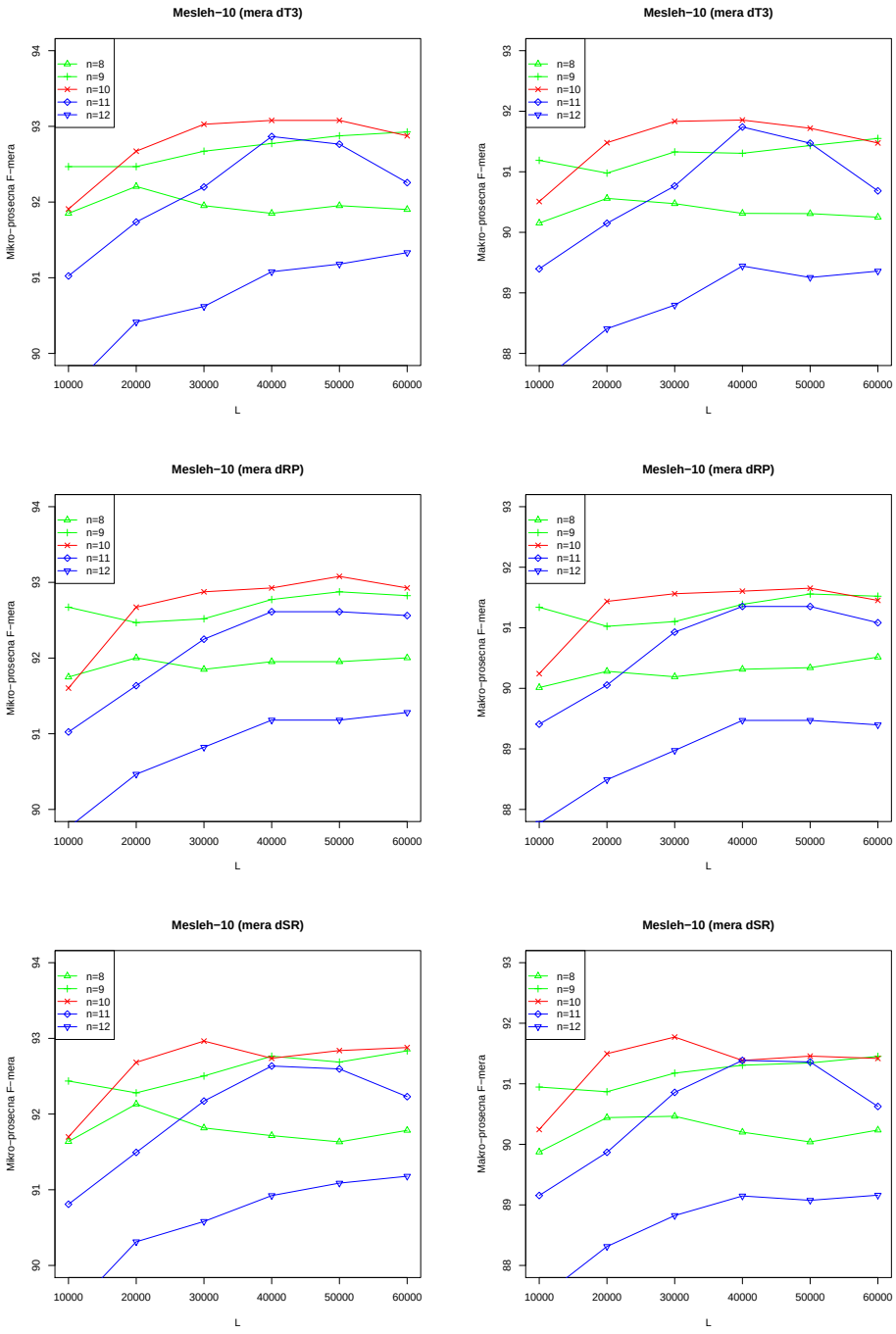


Slika 10: Poređenje osnovne varijante metode sa njenim modifikacijama na 20-Newsgroups korpusu, za $n = 7$ i mere različitosti $dT2$ i $dT3$, u terminima mikro- i makro-prosečne F-mere.

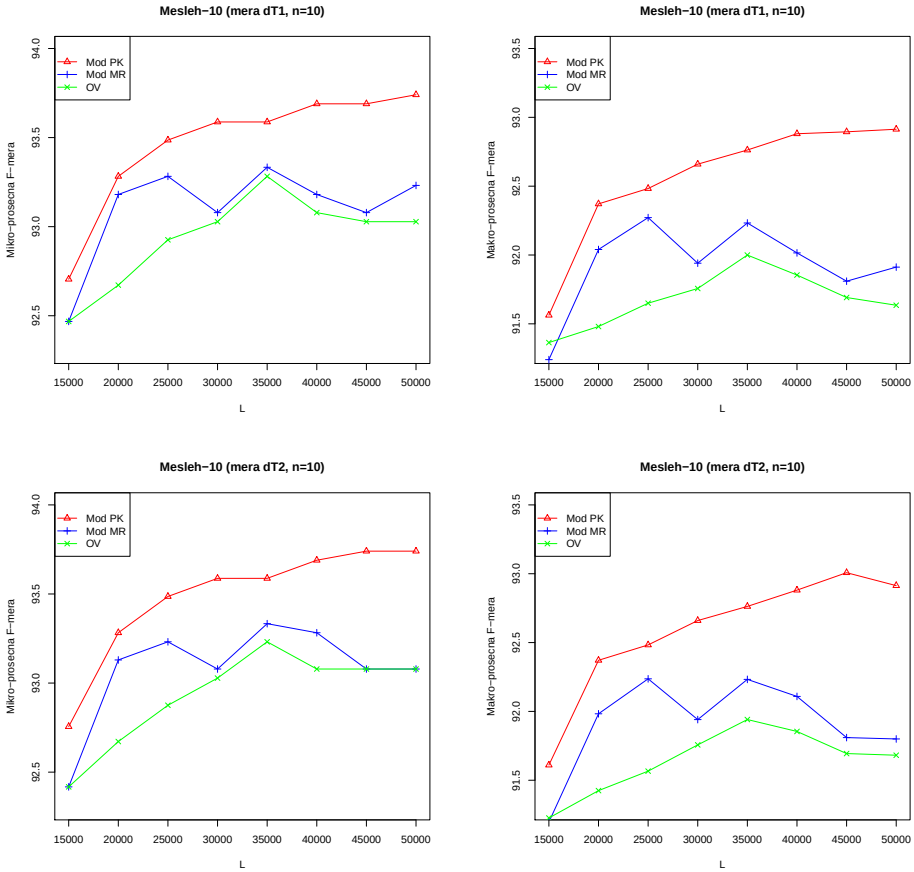
Mesleh-10



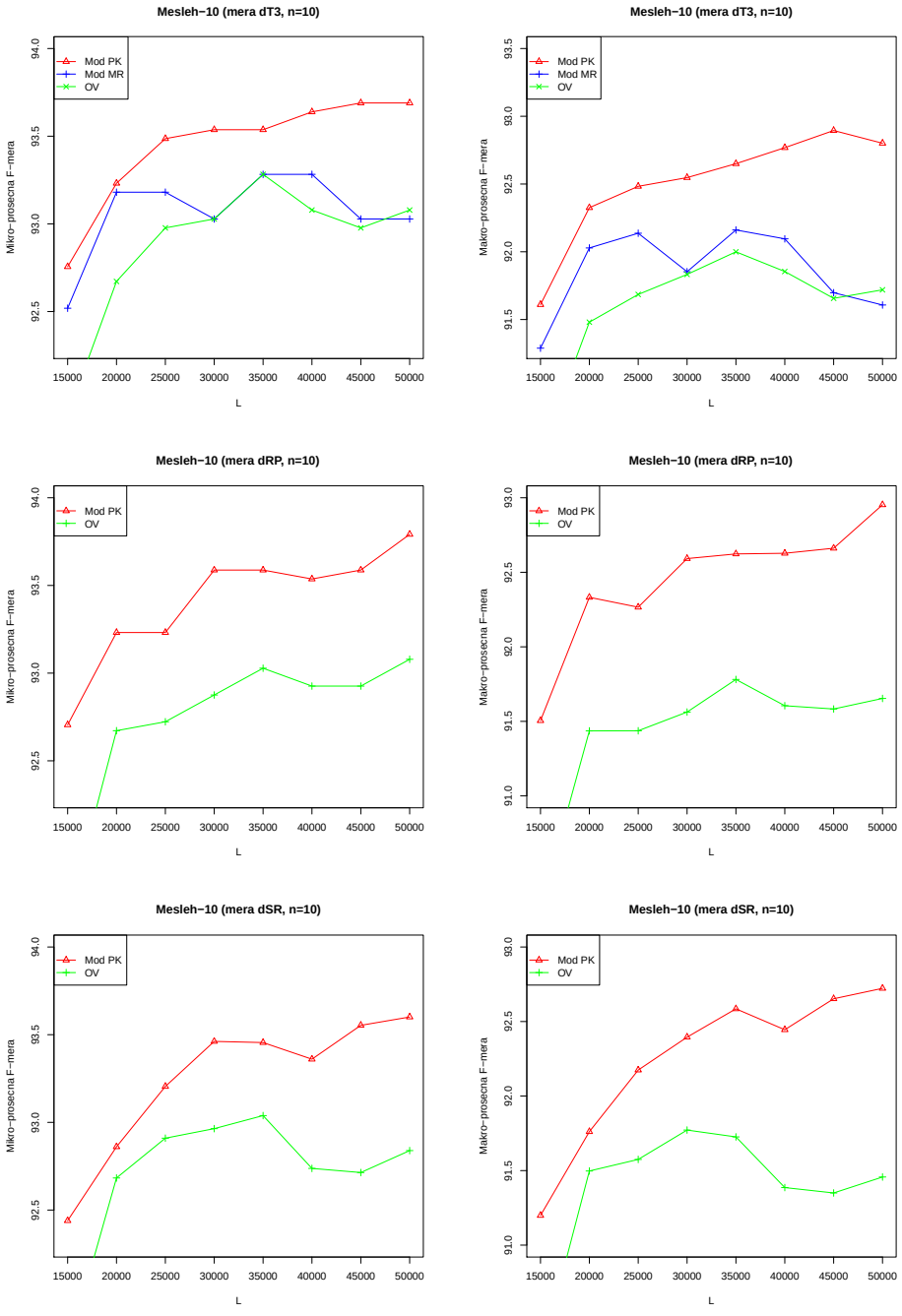
Slika 11: Mikro- i makro-prosečna F-mera za Mesleh-10 korpus, za različite vrednosti parametra n i mere različitosti $dT1$ i $dT2$.



Slika 12: Mikro- i makro-prosečna F-mera za Mesleh-10 korpus, za različite vrednosti parametra n i mere različitosti $dT3$, dRP i dSR.

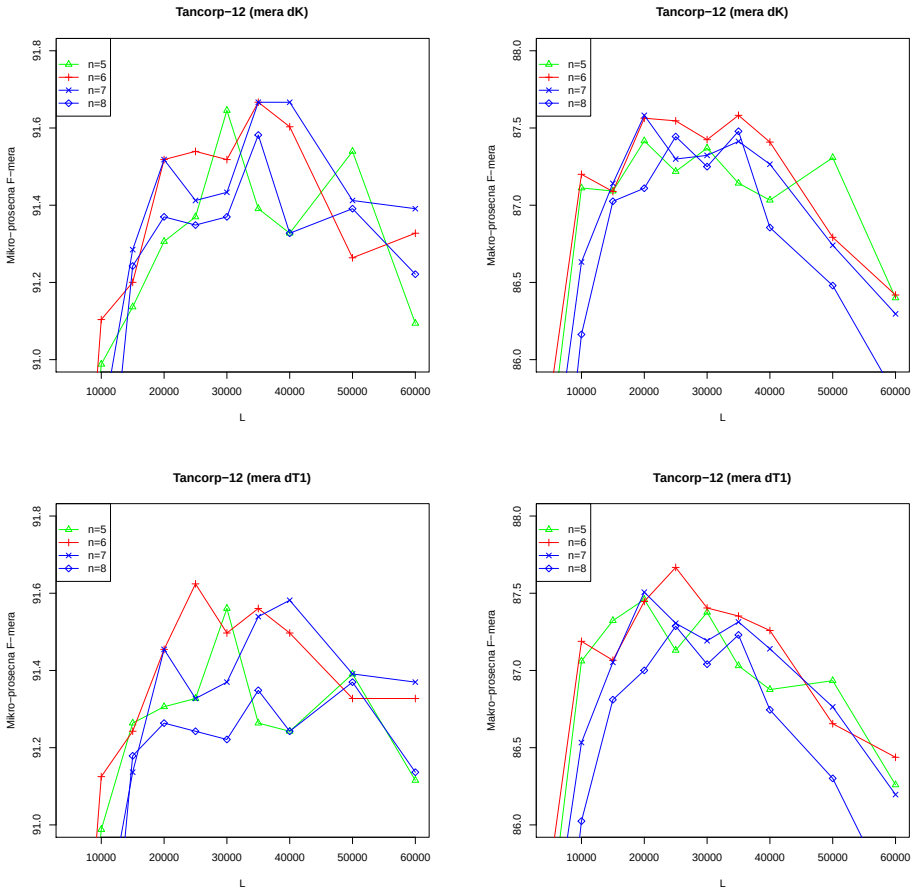


Slika 13: Poređenje osnovne varijante metode sa njenim modifikacijama na Mesleh-10 korpusu, za $n = 10$ i mere različitosti $dT1$ i $dT2$, u terminima mikro- i makro-prosečne F-mere.

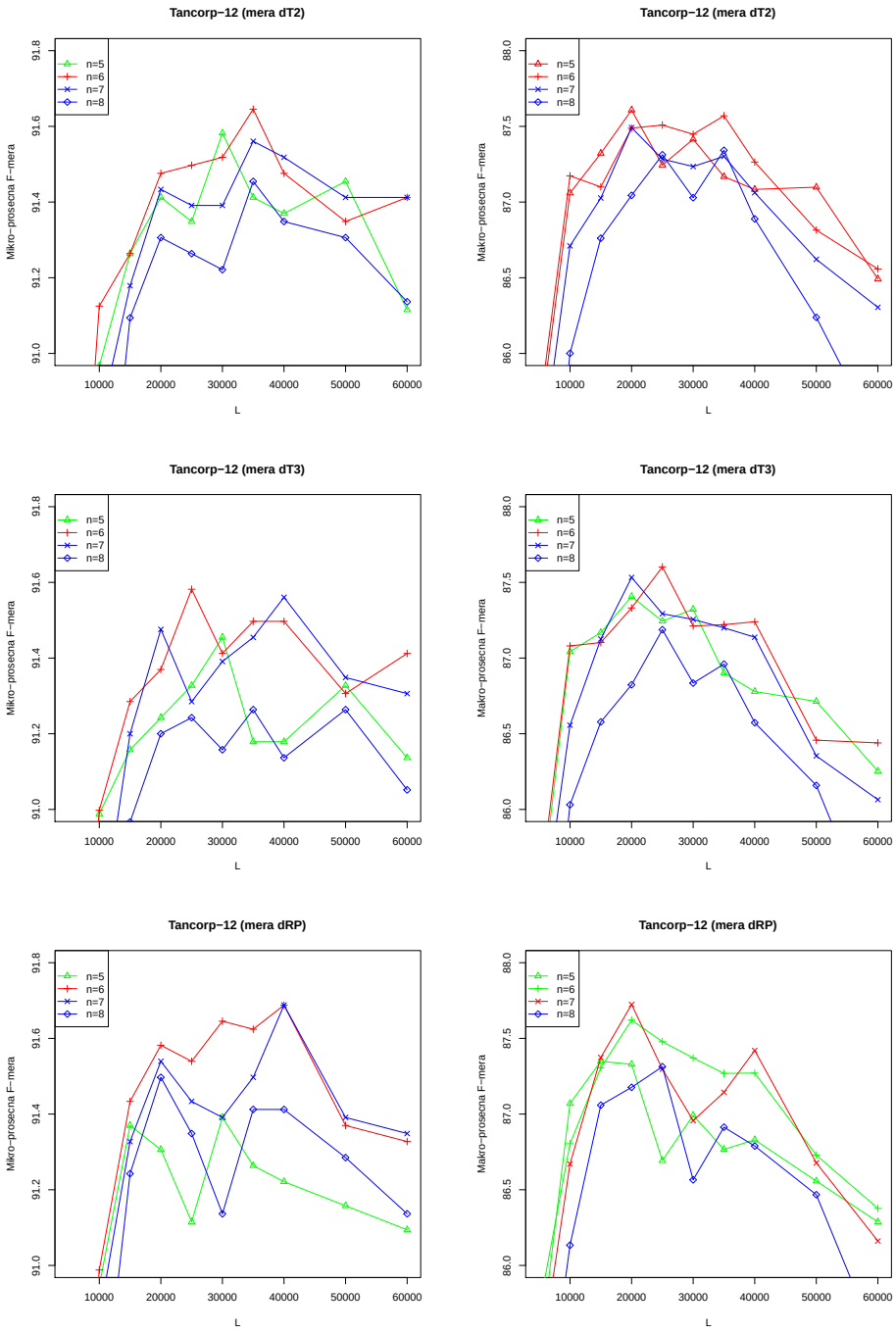


Slika 14: Poređenje osnovne varijante metode sa njenim modifikacijama na Mesleh-10 korpusu, za $n = 10$ i mere različitosti $dT3$, dRP i dSR, u terminima mikro- i makro-prosečne F-mere.

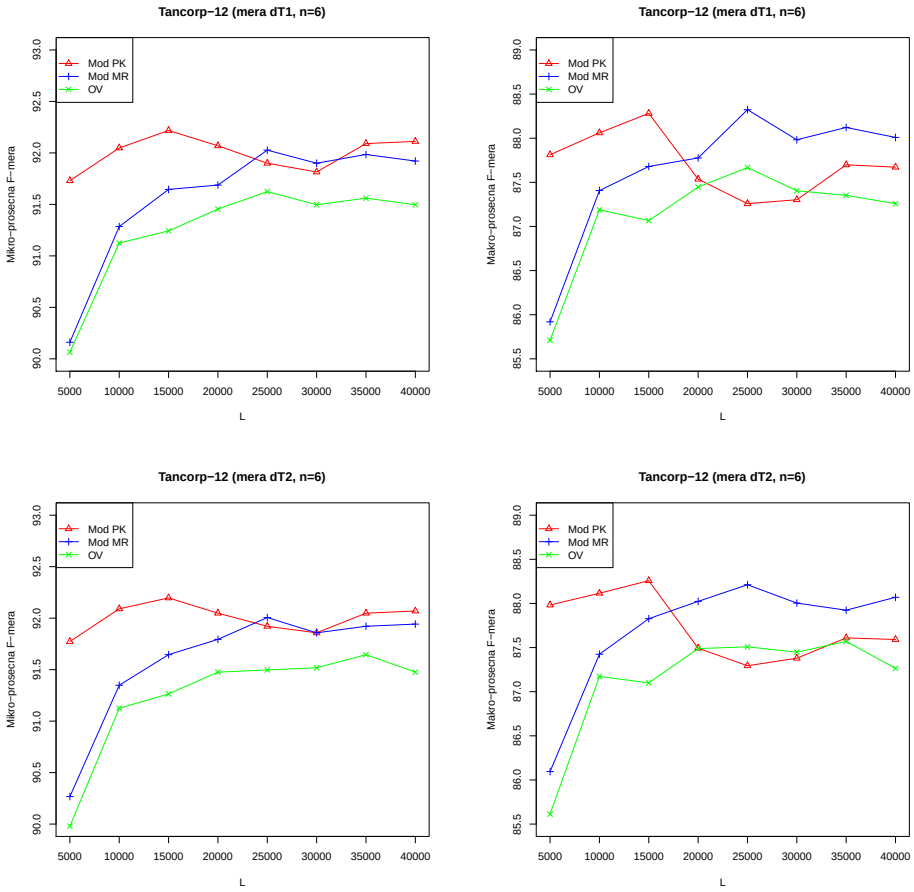
Tancorp-12



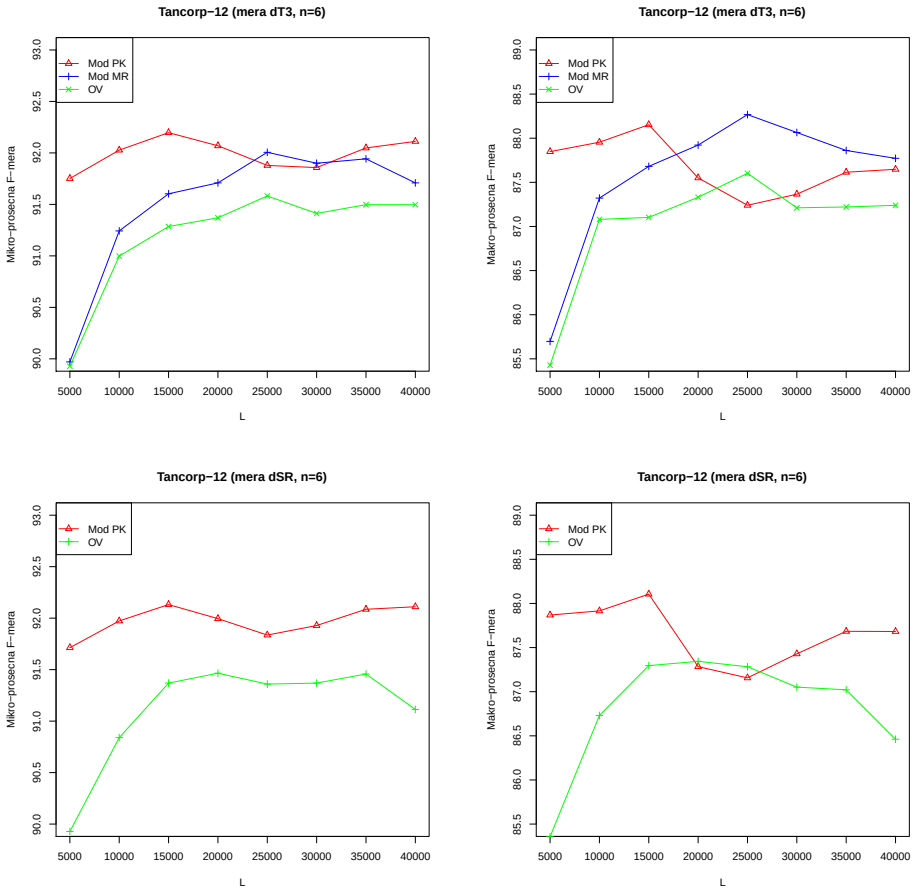
Slika 15: Mikro- i makro-prosečna F-mera za Tancorp korpus, za različite vrednosti parametra n i mere različitosti dK i $dT1$.



Slika 16: Mikro- i makro-prosečna F-mera za Tancorp korpus, za različite vrednosti parametra n i mere različitosti $dT2$, $dT3$ i dRP.



Slika 17: Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu, za $n = 6$ i mere različitosti $dT1$ i $dT2$, u terminima mikro- i makro-prosečne F-mere.



Slika 18: Poređenje osnovne varijante metode sa njenim modifikacijama na Tancorp-12 korpusu, za $n = 6$ i mere različitosti $dT3$ i dSR, u terminima mikro- i makro-prosečne F-mere.

Biografija autora

Jelena Graovac (rođ. Tomašević) rođena je 14. decembra 1979. godine u Prištini. Završila je OŠ "Dositej Obradović" u Prištini i Prvu prištinsku gimnaziju kao nosilac diploma "Vuk Karadžić" i đak generacije. Diplomirala je na Matematičkom fakultetu u Beogradu (smer Računarstvo i informatika) sa prosečnom ocenom 9,62. Tokom studija bila je nosilac stipendije Fondacija za razvoj naučno-istraživačkog podmlatka Ministarstva prosvete Republike Srbije, i stipendije kraljevine Norveške za 500 najboljih studenata u Srbiji. Školske 2004/2005. upisala je magistarske studije na Matematičkom fakultetu u Beogradu a 2008. godine odbranila je magistarsku tezu pod nazivom "XML baze podataka u upravljanju leksičkim resursima" (mentor prof. dr G. Pavlović-Lažetić).

Od 2004. godine Jelena Graovac zaposlena je na Matematičkom fakultetu kao asistent pripravnik, zatim saradnik u nastavi, a od 2008. kao asistent. Osnovne oblasti interesovanja su joj klasifikacija teksta, računarska obrada teksta i XML baze podataka. Učesnik je projekata "Automatsko rezonovanje i napredne obrade velikih količina podataka i teksta" i "Infrastruktura za elektronski podržano učenje u Srbiji" kojie finansira Ministarstvo nauke Republike Srbije. Objavila je veći broj naučnih radova i učestvovala na nekoliko međunarodnih i domaćih konferencija.

Udata je i ima dve ćerke.