SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1505

PRIMJENA JEZGRENIH METODA U KATEGORIZACIJI TEKSTA

Mislav Malenica

Zagreb, rujan 2004.

Zahvaljujem se prof. dr. sc. Bojani Dalbelo-Bašić
na usmjeravanju i ohrabrivanju,
dečkima iz laboratorija za Inteligentne sustave sa ZEMRIS-a na pomoći.
Hvala mojoj obitelji: mami, tati, Duji, Anti, Mariji i Juraju
na razumijevanju i potpori.

Posvećeno Sandi, bez koje sve ovo niti ne bi imalo smisla.

Sadržaj

Predgovor

Ovaj diplomski rad nastao je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave - ZEMRIS - Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu u okvirima projekta primjene informacijske tehnologije Ministarstva znanosti, obrazovanja i športa pod nazivom "Text mining system - Sustav za automatsko indeksiranje, kategorizaciju i semantičko pretraživanje teksta", voditelj projekta prof. dr.sc. Bojana Dalbelo Bašić. Svi eksperimenti rađeni su u Laboratoriju za Inteligentne sustave na ZEMRISU-u. Tijekom izrade rada surađivao sam s prof. dr.sc. Markom Tadićem, s Filozofskog fakulteta u Zagrebu (Odsjek za lingvistiku), na problemima pripreme i predprocesiranja podataka na hrvatskom jeziku. U eksperimentima su korišteni i rezultati automatske morfološke normalizacije teksta za hrvatski jezik, čiji je autor kolega dipl. ing. Jan Šnajder.

Rad zamišljen je kao polazna točka svima onima koji se nalaze na istom mjestu s kojega sam i sam krenuo prije godinu dana: izgubljen u moru literature i bez ideje kako dalje. Nastojao sam sve pojmove vezane uz kategorizaciju teksta sustavno uvesti i dobro ih definirati tako da i čitatelj s elementarnim znanjem o ovoj problematici ne bi trebao imati problema pri praćenju ovoga izlaganja. Onima s većim iskustvom na ovom području, zanimljiva će biti poglavlja o implementaciji sustava te poglavlja s rezultatima i vrednovanjem mjerenja. U ovom se radu po prvi put iznose rezultati kategorizacije teksta za bazu novinskih članaka Vjesnik, a ovo je ujedno i jedan od prvih pokušaja kategorizacije teksta na hrvatskom jeziku. Čitateljima koji nisu upoznati s pojmovima strojnog učenja ili još uvijek nisu sigurni koliko im je ova tema interesantna preostaje pogledati poglavlje s dodacima u kojem je na temelju jednostavnog primjera napravljen sažetak ovog rada.

1 Uvod

Internet i digitalni zapis učinili su dostupnima gomile informacija. Preplavljeni silnom količinom digitalnog teksta različitog porijekla, sadržaja i kvalitete nastojimo pronaći i usavršiti metode koje će omogućiti što lakše i uspješnije pretraživanje, filtriranje i upravljanje ovim informacijama. Kategorizacija teksta, metoda koja tekstu pridaje neku od unaprijed definiranih kategorija, jedan je od koraka u tim nastojanjima.

Ovaj se rad sastoji od dvije zaokružene cjeline. Prva cjelina daje sustavno prikazanu teorijsku podlogu i služi kao osnova za drugu, koja opisuje implementirani klasifikator teksta te prikazuje i vrednuje njegove rezultate.

2 Prikaz teksta

Kako bi sustav za dohvat informacija mogao pretražiti kolekciju dokumenata, dokumenti unutar toga sustava moraju biti predstavljeni na određeni način. Idealan prikaz trebao bi obuhvatiti sadržane informacije dokumenata što je točnije moguće, kako bi se dokumenti sličnog sadržaja mogli poistovjetiti, a različitog razlikovati (Viestam, 2001).

2.1 Načini prikaza

Čak i kada nam je tekst već pohranjen u računalu, najvjerojatnije to nije oblik prikladan većini algoritama strojnog učenja. Prvi nam je zadatak transformirati ga u prikaz pogodan i algoritmu učenja i postupku klasifikacije, jer se pokazalo da prikaz teksta ima vrlo velik utjecaj na rad klasifikatora, osobito na sposobnost generalizacije (Joachims, 2001). Osnovna podjela modela prikaza, u odnosu na razinu na kojoj se obrađuje tekst je (Joachims, 2001):

1. nivo fragmenata riječi: dekompozicija riječi, morfološke informacije
2. nivo riječi: riječi, leksičke informacije
3. nivo skupina riječi: fraze, sintaksne informacije
4. semantički nivo: značenje teksta
5. pragmatički nivo: značenje teksta s obzirom na kontekst i situaciju

U osnovi, što je viši nivo analize teksta, više se informacija prikupi, međutim time se strahovito povećava složenost automatske ekstrakcije takvih podataka.

2.2 Automatsko indeksiranje

Prije nego što započnem s objašnjenjem razlike među metodama prikaza teksta, čini mi se neophodnim objasniti pojam automatskog indeksiranja, jer su sve gornje metode više ili manje vezane uz njega.

Krećemo od pretpostavke da prisutnost ili odsutnost riječi ili izraza (jedna riječ ili kombinacija riječi) ukazuje na temu teksta (Karlgren,2000). Možemo odmah

dati definiciju automatskog indeksiranja: automatski indeks može se definirati kao grupa riječi ili fraza koje je računalo odabralo iz teksta s namjerom da olakša pronalaženje i dohvat informacija te da pruži uvid o vrsti informacija koje se nalaze u tekstu[1].

Još u šesnaestom stoljeću postojala je ideja o indeksima kao listama ključnih riječi s pokazivačima na dokumente[2]. Osnovna ideja bila je da se tražene informacije pronalaze izborom dokumenata koji su se nalazili na popisu prigodne ključne riječi. Prije računala, ovaj su vremenski zahtjevan posao ljudi obavljali ručno. Prednost takvog postupka pred računalnom obradom je u kvaliteti izabranih ključnih riječi, jer su ljudi sposobni u potpunosti shvatiti značenje teksta. Međutim, ovakav posao iznimno je dosadan i mukotrpan te zahtjeva mnogo vremena, a s obzirom na obim dostupnog teksta, više se niti ne može obavljati ručno.

Kategorizacija[3] ovisi o subjektivnom mišljenju osobe koja ju obavlja, te je teško zamislivo da ta osoba vremenom ostane dosljedna. Konzistentnost koja je ključna pri opisivanju kategorije izostaje i kad jedna osoba obavlja indeksiranje, a naročito kad ga obavlja više njih. Kako bi izbjegli varijacije kod ručnog indeksiranja, indeksi se ograničavaju na određeni skup dobro definiranih izraza; time se povećava dosljednost, ali na račun fleksibilnosti. Ne treba zaboraviti ni činjenicu da definicije izraza zastarijevaju puno brže od samih izraza (Karlgren,2000). Vidjevši da i ručno indeksiranje ima svojih nedostataka, odustalo se od namjere da se računalima pokušaju imitirati ručne metode. Automatske metode više se ne zasnivaju na indeksiranju pomoću unaprijed definiranog skupa izraza, već, uglavnom, koriste izraze pronađene u tekstu. Tu dolazi do izražaja snaga računala koja su sposobna statistički obraditi velike količine podataka u potrazi za odgovarajućim indeksima.

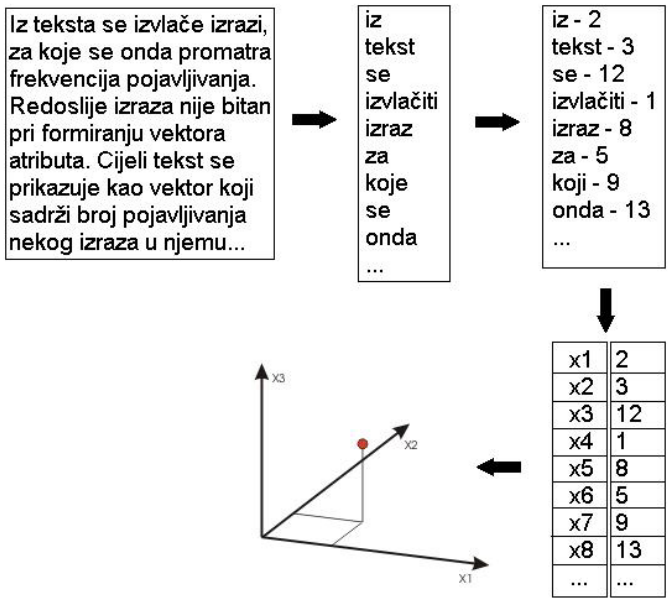
---

[1] L.L. Earl, (1970), Experiments and Automatic Extracting and Indexing, *Information Storage and Retrieval*, 6, pp.313-334, Pergamon Press, via (Viestam,2001)

[2] H.B. Wheatley, (1979), What is an Index? A Few Notes on Indexes and Indexers, Longmans Green / co., London, via (Viestam,2001)

[3] Kategorizacija ili klasifikacija teksta je postupak određivanja pripadnosti teksta određenoj klasi ili kategoriji tekstova.

2.2.1 Nivo riječi



Slika 2.1 Prikaz postupka indeksiranja

Osnovna je zamisao ogoliti tekst do osnovnih gradivnih jedinica – riječi, te svakoj riječi pridružiti frekvenciju pojavljivanja (broj pojavljivanja određene riječi u tekstu). Primjer postupka prikazan je na slici 2.1.

Polazimo od pretpostavke da je za određivanje kategorije dokumenta, odnosno za klasifikaciju, dostatna količina informacije pohranjena u činjenici da se određene riječi pojavljuju ili ne pojavljuju u tekstu, bez obzira na njihov redoslijed i ulogu. Iako postoje homonimi i sinonimi, a kontekst mijenja značenje nekih izraza, ti utjecaji se na ovoj razini analize teksta zanemaruju. Jasno je da se ovakvim prikazom gubi dio informacije o dokumentu, međutim, ni mnogo složenije metode analize teksta ne postižu mnogo bolje rezultate.

Iako ekspresivnije metode mogu obuhvatiti više informacija o tekstu, njihova veća složenost smanjuje kvalitetu statističkih modela izgrađenih nad njima. Stoga se prikaz na nivou riječi čini kao dobar kompromis između ekspresivnosti i složenosti (Joachims, 2001).

2.2.2 Nivo fragmenata riječi

Umjesto cijelih riječi, za indekse se koriste n-grami - nizovi slova, duljine n iz tih riječi. Npr. ako se radi o riječi "slovo", 4-grami su: "_slo", "slov", "lovo", "ovo_".

Prednosti ove metode su: zaštita od pogreški u pisanju, modeliranje sličnosti[4] među riječima, npr. riječi «model» i «modelar» imaju zajedničku većinu n-grama, te linearna složenost.

2.2.3 Nivo skupina riječi

Želimo li izvući što veću količinu informacije, moramo promatrati i neke sintaksne zakonitosti među riječima. Na taj smo način sposobni registrirati neke složene izraze (uglavnom fraze ili tehničke izraze). Ovu metodu promatramo sa stajališta da nam više informacije o kategoriji dokumenta nudi izraz "metoda potpornih vektora" koji promatramo kao cjelinu, nego kao skup od tri nepovezane riječi.

Detekciju ovakvih složenih izraza vršimo usporedbom s gotovom listom složenih izraza ili statističkom metodom pri kojoj se gleda učestalost zajedničkog pojavljivanja riječi iz jedne takve grupe (Joachims, 2001).

Više se puta pokazalo kako detekcija složenih izraza tek marginalno doprinosi procesu dohvata informacija, te kako se trud koji je potrebno uložiti u takvu implementaciju ne isplati[5].

---

[4] Međutim, ova može i zavarati, jer postoje mnoge riječi kod kojih bi sličnost definirana na ovaj način bila velika, a ipak su potpuno različitog značenja, npr. "poglavar" i "poglavlje". Očito je da se radi o sličnosti definiranoj na leksičkoj osnovi.

[5] K. Sparck Jones, (1997), What is the role of NLP in Text Retrieval, via (Karlgren,2000).

2.3 Semantički nivo

Različite su metode obuhvaćanja semantičkog značenja teksta. Jedna je od njih latentno semantičko indeksiranje[6] kojim se automatski generiraju semantičke kategorije bazirane na prikazu na nivou riječi. Druga je metoda uz pomoć predikatne logike i semantičkih mreža kojima se mogu opisati veoma upotrebljivi jezici, no kako ovaj postupak zahtjeva ručne translacije dokumenata, neupotrebljiv je za većinu aplikacija.

Za optimalan rad klasifikatora teksta potrebno je da on dovoljno dobro obuhvati i semantičko značenje teksta, međutim, još uvijek ne postoje automatske metode koje bi transformirale slobodan tekst u oblik koji bi mogli koristiti algoritmi učenja (Joachims,2001).

2.4 Problematika prirodnog jezika

Osnovni problem pri radu s prirodnim jezikom je taj što kontekst u kojemu je nešto izrečeno ima vrlo veliki utjecaj na značenje izrečenog. Obuhvatiti kontekst prirodnog jezika znači opisati njegovu strukturu. Bogatstvo prirodnog jezika onemogućava stvaranje sustava koji bi ga u potpunosti obuhvatio. Za sada se čini nemoguće pristupiti opisu prirodnog jezika bez postavljanja nekih odredimo li da rečenicom ili dijelom teksta možemo obavljati samo sljedeće radnje:

- opisati drugima (istinito ili lažno) stanje stvari
- pokušavati navesti druge da nešto učine
- obvezati se da ćemo nešto učiniti
- objaviti promjene u svijetu oko nas, a koje su rezultat našega djelovanja
- opisati osobne osjećaje i stavove,

[6] Latentno semantičko indeksiranje definirano je i detaljno objašnjeno u (Deerwester,1990).

još uvijek postoji mnogo toga što moramo pretpostaviti, npr. pretpostavke o namjerama govornika :

- trebao bi biti informativan koliko nam je potrebno i ne više od toga
- ne bi trebao izreći ono što smatra netočnim
- ne bi trebao izreći ono za što mu nedostaje dokaza
- trebao bi govoriti o relevantnim stvarima
- trebao bi izbjegavati nedefinirane i nejasne izraze
- trebao bi izbjegavati dvosmislene izraze
- trebao bi govoriti jezgrovito

Dok pokušavamo razložiti jezik na njegove komponente, opisati ga pomoću struktura, obuhvatiti pravilima neprestano nailazimo na nove probleme. Čini se kako je struktura jezika dinamična i teško da je možemo opisati nizom pravila.

Krenemo li putem isključivanja lingvističke metode te baziranja samo na golim činjenicama poput: frekvencija pojavljivanja riječi ili izraza bilo u dokumentima, rečenicama, paragrafima ili unutar neke udaljenosti, mogli bi izgraditi iznimno složene sustave. Ipak, teško bi bilo složiti se s izjavom da oni mogu u potpunosti obuhvatiti značenje dokumenta koje bi sadržavalo temu, sadržaj, svrhu, primjenu ili veze s drugim dokumentima. Pitanje je kako statistički obuhvatiti i osjećaj da znamo mnogo o onome što će govornik reći prije nego što on išta i kaže (Blair,1992)? No, isto tako, pogrešno bi bilo tvrditi kako navedene činjenice o tekstu nisu u korelaciji sa značenjem dokumenta. Možda je pravo pitanje: «Trebamo li mi zapravo potpuno značenje dokumenta?"

3 Odabir atributa

Osnovni prikaz koji nastaje indeksiranjem riječi iz teksta sadrži mnoge nebitne i neprimjerene atribute. Dva su osnovna motiva za redukciju skupa atributa (Joachims,2001): prvi je zaštita od prenaučenosti, a drugi leži u činjenici da mnogi algoritmi ne mogu podnijeti rad u tako visokodimenzionalnom prostoru kojega nameće osnovni prikaz teksta.

3.1 Svojstva zadaće klasifikacije teksta

Kako bi mogli što učinkovitije pristupiti procesu odabira atributa, važno je dobro razumjeti problematiku klasifikacije teksta.

Prva karakteristika koju možemo uočiti je velika dimenzionalnost prostora atributa. Koristimo li svaku riječ iz skupa od otprilike 10000 dokumenata, dobit ćemo prostor atributa dimenzije oko 25000 – 250000, ovisno o prosječnoj duljini dokumenta[7].

Iako postoji veliki broj mogućih atributa za određeni skup, svaki od dokumenata sadrži samo mali broj različitih riječi[8]. Dakle, vektori koji predstavljaju dokumente vrlo su rijetko popunjeni (eng. sparse).

Želimo li izbjeći ovako visoku dimenzionalnost, mogli bismo pokušati odbaciti sve nevažne atribute. Međutim, preklapanja među dokumentima uglavnom su vrlo mala, te sva veća rezanja na skupu atributa završavaju gubitkom informacije. Ovo je

[7] Neovisno o vrsti teksta, postoji čvrsta veza između veličine dokumenta i različitih riječi koje se pojavljuju u njemu. Ta veza naziva se Heapsov zakon, a formulirana je izrazom $V=k*s^{\beta}$, gdje su k (obično između 10 i 100) i β (obično između 0.4 i 0.6) konstante ovisne tipu teksta, a s broj dokumenata. (H.S. Heaps, (1978), Information Retrieval: Computational and Theoretical Aspects, Academic Press, New York) via (Joachims,2001)

[8] Dokumenti iz Reutersovog skupa imaju prosječnu duljinu od 152 riječi, među kojima su prosječno 74 različite (Joachims,2001).

posljedica činjenice što prirodni jezik dopušta da istu stvar kažemo na različite načine[9].

Većina dokumenata posjeduje više od jednog izraza za opisivanje klase. Dakle, uklonimo li one koje nose najviše informacija o dotičnoj klasi, opet će se na temelju preostalih moći izvršiti određena klasifikacija. To znači da svaku klasu karakterizira više atributa sa sličnom raspodjelom te da su vektori dokumenata redundantni s obzirom na zadaću klasifikacije.

Frekvencije izraza prirodnog jezika ponašaju se vrlo stabilno. Prvi je pokušaj opisivanja toga ponašanja Zipfov zakon[10] koji kaže da ukoliko izraze poredamo prema njihovoj frekvenciji, onda je frekvencija r-tog izraza 1/r puta frekvencija najfrekventnijeg izraza. Formula je dana izrazom (3.1). To bi značilo da se samo mali broj izraza pojavljuje jako često[11], dok se gotovo polovica izraza može pojaviti i samo jednom. Zipfov zakon opisuje osnovni princip ponašanja, no pokazalo se da Mandelbrotova raspodjela, izraz (3.2), znatno bolje odgovara eksperimentalnim podacima[12]. Oznaka r predstavlja redno mjesto izraza u poretku prema frekvencijama, dok su k, c i ϕ konstante. Iz izraza je vidljivo kako je Zipfova distribucija zapravo specijalan slučaj Mandelbrotove raspodjele.

$$TF = \frac{\max\_frekvencija}{r} \tag{3.1}$$

$$TF = \frac{c}{(k+r)^{\phi}} \tag{3.2}$$

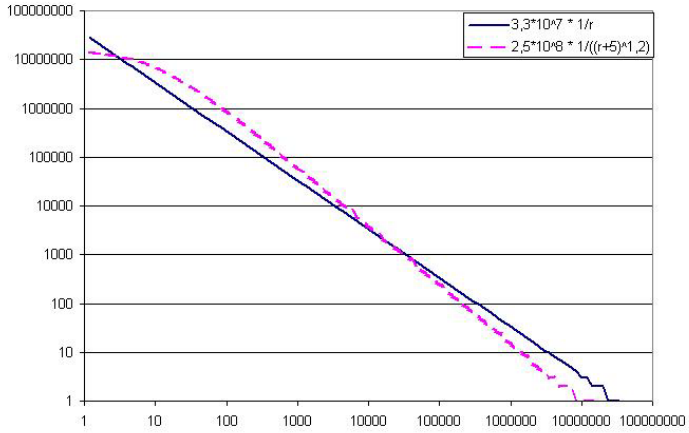
---

[9] Upotrebljavajući različite rečenične konstrukcije te različite riječi.

[10] G.K. Zipf, (1949), Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, Addison-Wesley, Cambridge, MA, USA, via (Joachims,2001).

[11] Za rječnik od 10000 riječi, 100 najfrekventnijih riječi se pojavljuju u oko 50% slučajeva (Joachims,2001).

[12] M.D. Araujo, G.Navarro, N. Ziviani, (1997), Large text searching allowing errors. In Baeza Yates, R., editor, Proceedings of the 4th South American Workshop on String Processing, pp. 2-20, Valparaiso, Chile, Carleton University Press. B. Mandelbrot, (1959), A note on a class of skew distribution functions: Analysis and critique of a paper by H.A.Simon, Information and Control, 2(1), pp. 90-99. G.A. Miller, E.B. Newman, E.A. Friedman, (1958), Length-frequency statistics for written English, Information and Control, 1(4), pp.470-389. via (Joachims, 2001).



Slika 3.1 Primjeri Zipfove i Mandelbrotove raspodjele

Na slici 3.1 prikazani su primjeri koji pokazuju oblik Zipfove i Mandelbrotove raspodjele.

Dva su pristupa procesu odabira atributa:

1. **odabir podprostora atributa** – odabire se podskup početnog skupa atributa
2. **konstrukcija atributa** – atributi nastaju konstrukcijom iz atributa početnog skupa

3.2 Odabir podprostora atributa

Odabir atributa vršimo na temelju određenog znanja zbog kojeg smo u stanju usporediti atribute te odabrati najbolje među njima. To znanje može biti lingvističke ili statističke prirode. U ove metode ubrajamo:

1. eliminaciju stop riječi
2. frekvenciju dokumenata - DF
3. informacijsku dobit - IG
4. uzajamnu informaciju
5. χ^2 test

3.2.1 Eliminacija stop riječi

Jedan od načina eliminiranja irelevantnih atributa iz skupa je eliminacija stop riječi. Stop riječi su one riječi koje ne nose informaciju ni o jednoj kategoriji. To su najčešće veznici, prijedlozi i prilozi. Karakterizira ih visoka frekvencija pojavljivanja. Primjeri za engleski jezik su: *and, if, or, when, the...*, a za hrvatski: *i, na, u, već, ali,...*. Stop riječi filtriramo na temelju pripremljene liste ili na temelju učestalosti pojavljivanja (Forman,2003).

3.2.2 Frekvencija dokumenata - DF

Frekvencija dokumenata je broj dokumenata iz kolekcije u kojima se pojavio određeni izraz. Izračuna se frekvencija dokumenata za sve izraze, a zatim odbacuju izrazi, odnosno atributi, čija frekvencija dokumenata ne prelazi određeni prag. Čak i za niske pragove, dimenzionalnost prostora značajno se reducira.

Ova metoda polazi od pretpostavke da rijetki izrazi nisu dovoljno informativni ili nisu dovoljno pouzdani[13] kako bi se uzeli u obzir pri klasifikaciji.

[13] Naime, šum u tekstu (pogrešno napisane riječi) pojavljuje se u obliku niskofrekventnih riječi. Stoga, odbacivanjem niskofrekventnih atributa, smanjujemo dimenzionalnost i odstranjujemo šum.

Ovo je metoda koja se najlakše implementira, upotrebljiva je i na velikim skupovima. Složenost je gotovo linearna s obzirom na broj dokumenata. Međutim, nije efikasna kod agresivne redukcije zbog toga što se smatra kako su i izrazi s nižim frekvencijama zapravo vrlo informativni.

3.2.3 Informacijska dobit - IG

Informacijska dobit mjeri količinu informacije o kategoriji kojoj pripada dokument, a koju nam donosi spoznaja o prisutnosti određenog izraza u tom dokumentu. Ako imamo m različitih kategorija označenih s $c_1,\ldots,c_m$, onda je izraz kojim se definira informacijska dobit dan s izrazom (3.3).

$$IG(t) = -\sum_{i=1}^{m} p(c_i)\log(p(c_i)) + p(t)\sum_{i=1}^{m} p(c_i \mid t)\log(p(c_i \mid t)) + p(\bar{t})\sum_{i=1}^{m} p(c_i \mid \bar{t})\log(p(c_i \mid \bar{t})) \qquad (3.3)$$

Za svaki atribut izračunamo informacijsku dobit te odbacimo sve one atribute čija je informacijska dobit niža od nekog predodređenog praga.

Kako bi dobili IG, moramo prvo izračunati pripadajuće uvjetne vjerojatnosti, a zatim entropiju. Izračun uvjetnih vjerojatnosti karakterizira vremenska složenost O(*N*) i prostorna složenost O(*VN*), gdje je N broj dokumenata, a V broj atributa. Izračun entropije ima vremensku složenost O(*Vm*) (Yang,1997).

3.2.4 Uzajamna informacija

Uzajamna informacija definira se izrazom (3.4).

$$I(t,c) = \log\frac{P(t \wedge c)}{P(t) \times P(c)} \qquad (3.4)$$

Oznakom *t* predstavljamo izraz, a oznakom *c* kategoriju. Vrijednost I(t,c) jednaka je nuli kada su *t* i *c* nezavisni. Želimo li koristiti ovu mjeru pri izboru atributa (izraza), moramo dobiti vrijednost određenog izraza uzimajući u obzir sve kategorije. To, uglavnom, radimo tako da izračunamo ili prosječan rezultat ovog izraza po kategorijama (3.5) ili da izraz predstavimo maksimalnom vrijednošću koju postiže na jednoj od kategorija (3.6).

$$I_{avg}(t) = \sum_{i=1}^{m} P(c_i) I(t,c_i) \qquad (3.5)$$

$$I_{\max}(t) = \max_{i=1}^{m} \{I(t,c_i)\} \qquad (3.6)$$

Vremenska složenost izračuna uzajamne informacije iznosi O(Vm), slično kao i kod informacijske dobiti.

Nedostatak ovog kriterija leži u činjenici što daje prednost rijetkim izrazima. To se možda može najbolje uočiti na primjeru izraza (3.7), koji je ekvivalentan izrazu (3.4), gdje je vidljivo kako kod izraza koji imaju jednake uvjetne vjerojatnosti viši rezultat postiže onaj koji se rjeđe pojavljuje. Stoga možemo zaključiti da rezultati nisu usporedivi za izraze koji se jako razlikuju u frekvencijama pojavljivanja.

$$I(t,c) = \log P_r(t \mid c) - \log P_r(t) \tag{3.7}$$

3.2.5 χ^2 test

χ^2 test je statistički test koji mjeri odstupanje od očekivane raspodjele, ako pretpostavimo da je pojavljivanje izraza neovisno o kategoriji (Forman,2003). Jednostavnije rečeno, χ^2 test mjeri nedostatak nezavisnosti između izraza *t* i kategorije *c*. Ovaj test dan je izrazom (3.8), gdje *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 članaka kada se niti *t* niti *c* nisu pojavili te je *N* broj dokumenata u skupu.

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{3.8}$$

Kada χ^2 test upotrebljavamo za odabir atributa, moramo na neki način sumirati rezultate po kategorijama. Izračun prosječnog odstupanja od raspodjele (3.9) ili uzimanje maksimalnog odstupanja (3.10) dva su primjera toga postupka.

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i)\chi^2(t,c_i) \tag{3.9}$$

$$\chi^2_{\max}(t) = \max_{i=1}^{m} \{\chi^2(t,c_i)\} \tag{3.10}$$

Računanje χ^2 test ima kvadratnu složenost, slično kao i informacijska dobit te uzajamna informacija. Rezultati χ^2 testa međusobno su usporedivi, jer se radi s normaliziranim vrijednostima. Međutim, kao statistički test poznat je po tome što je nepouzdan za male vrijednosti koje su česte[14] pri klasifikaciji teksta.

[14] Male vrijednosti posljedica su izraza koji se rijetko pojavljuju, a ponekad i zbog toga što se koncept pokušava prikazati premalim brojem pozitivnih primjera (Forman,2003).

3.3 Konstrukcija atributa

Do sada smo govorili o metodama koje među ponuđenim atributima biraju one za koje smatraju da su najvrjedniji u postupku klasifikacije. Drugi pristup bio bi pokušati od ponuđenog skupa atributa stvoriti novi skup kao produkt atributa starog skupa. U ovu skupinu metoda ubrajamo sljedeće:

1. svođenje na korijen riječi
2. lematizaciju
3. tezaurus
4. latentno semantičko indeksiranje
5. konceptno indeksiranje

3.3.1 Svođenje na korijen riječi

Ovom metodom smanjujemo veličinu indeksa time što poistovjećujemo izraze s istim korijenom[15]. Primjer je korijen *connect* s kojim poistovjećujemo riječi: *connected*, *connecting*, *connection* i *connections*. Smisao svođenja na korijen riječi je u tome da se više leksički različitih riječi predstavi izrazom koji označava koncept koji sve te riječi predstavljaju. Odmah se uočava i nuspojava, jer je netočno tvrditi kako riječi sa istim korijenom uvijek ukazuju na isti koncept ako se zna da i riječ sama po sebi dobiva svoje puno značenje tek u kontekstu (Viestam,2001).

Jedan od najčešće korištenih algoritama za određivanje korijena riječi je onaj kojeg je dao Porter (Porter,1980), a koji koristi kontekstno osjetljiva pravila za odbacivanje sufiksa. Porterov algoritam razmjerno je jednostavan, ali, nažalost, primjenjiv samo za engleski jezik koji je ionako morfološki vrlo siromašan (Karlgren,2000).

[15] Korijen riječi je ono što ostane nakon što riječi odstranimo prefiks i sufiks (ukoliko ih riječ ima).

Korisnost ove metode ne dolazi toliko do izražaja kod primjene na engleski jezik, no kod morfološki bogatijih jezika pokazana su značajnija poboljšanja[16] u radu.

3.3.2 Lematizacija

Lematizacija je proces sličan procesu određivanju korijena riječi. Razlika je u tome što se ovdje, umjesto na korijen riječi, riječ svodi na njen osnovni oblik - lemu. Ovaj je proces dosta teži od procesa određivanja korijena riječi, jer riječ moramo točno identificirati (Viestam,2001).

3.3.3 Tezaurus

Dimenzionalnost prostora smanjujemo pronalazeći, na osnovi tezaurusa, sinonime i gotovo sinonime te ih izjednačavamo, odnosno zamjenjujemo predstavnikom skupine. U svojoj osnovnoj formi, ovakav se tezaurus sastoji od liste riječi bitnih za danu domenu, a svaka od tih riječi vuče sa sobom skup riječi istog ili sličnog značenja (Baeza-Yates,1999). Ova će metoda najbolje rezultate dati ako se radi o specifičnoj domeni ili s kontroliranim rječnikom (Viestam,2001).

Tezaurus može sadržavati različite relacije među riječima. Osim sinonima koje vodimo pod relacijom ekvivalencije, tu ubrajamo i relacije poput *općenitije od* te *specifičnije od* (Joachims,2001).

3.3.4 Latentno semantičko indeksiranje

Dohvat informacija obično se izvodi usporedbom pojmova iz dokumenata s pojmovima iz upita. Međutim, leksičke metode mogu biti neprecizne kada se radi s korisničkim upitima, jer postoji mnogo načina na koje se može predočiti željeni koncept (sinonimi). Stoga pojmovi iz upita ne moraju odgovarati pojmovima iz nevažnih dokumenata. Također, mnogi pojmovi imaju višestruka značenja

[16] Od navedenih stranih jezika u (Karlgren,2000) nama je najzanimljiviji slovenski jezik za kojeg su rezultati prezentirani u (M. Popovic, P. Willett, The effectiveness of stemming for natural-language access to Slovene textual data. JASIS, 43 (5), pp. 384-390, 1992).

(polisemija), tako da pojmovi iz upita mogu doslovno odgovarati pojmovima iz nerelevantnih dokumenata. Latentno semantičko indeksiranje nastoji zaobići ove probleme izbjegavajući direktan rad s riječima iz dokumenata time što koristi statistički dobivene indicije o konceptima koje te riječi predstavljaju(Berry,2003).

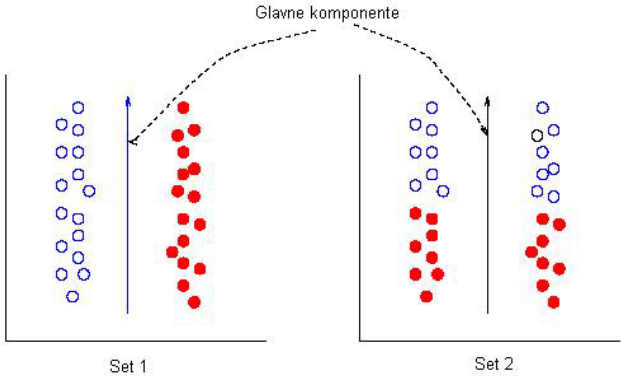
Latentno semantičko indeksiranje provodi preslikavanje vektora karakteristika u podprostor, smanjene dimenzionalnosti, koristeći se metodom dekompozicije singularnih vrijednosti (SVD). Izračunava se ortogonalna transformacija koordinatnog sustava, a nove koordinate odgovaraju novim karakteristikama. Odabire se samo *s* najvećih singularnih vrijednosti, stoga rezultirajući podprostor ima manju dimenziju uz najmanje moguće odstupanje[17]. Broj singularnih vrijednosti *s* koje treba odabrati kako bi se dobili najbolji rezultati je u početku nepoznat, a do njega se može doći krosvalidacijom (Joachims,2001).

LSI metoda dobro se nosi s problemom sinonima, jer će ih prilikom preslikavanja svrstati jedne do drugih. Međutim, problem polisemije se ne može riješiti ovom metodom, jer je svaki pojam predstavljen kao jedna točka u prostoru. Pojmovi s više značenja bit će predstavljeni prosječnom vrijednošću tih značenja koja mogu biti vrlo različita (Deerwester,1990).

3.3.5 Konceptno indeksiranje

Metode za dohvat informacija bazirane na redukciji dimenzionalnosti, poput latentnog semantičkog indeksiranja, pokazale su da poboljšavaju kvalitetu dohvaćenih informacija, jer uspijevaju obuhvatiti i prikrivena značenja riječi. Međutim, veliki računalski zahtjevi LSI metode te njena nemogućnost da u nadziranom okruženju vrši redukciju dimenzionalnosti, ograničavaju joj primjenjivost (Karypis,2000b).

[17] Odstupanje se odnosi na kvadratnu pogrešku.



Slika 3.2 Problem LSI metode pri klasifikaciji podataka

LSI metoda nije u stanju iskoristiti informaciju o kategoriji pojedinih dokumenata u postupku redukcije dimenzionalnosti pa ovisno o strukturi skupa podataka, efekti mogu jako varirati. Na slici 3.2 vidljivo je da bi ovakva metoda na prvom skupu rezultirala najgorim, a na skupu 2 najboljim mogućim slučajem.

Konceptno indeksiranje zasniva se na određivanju grupa sličnih dokumenata na temelju kojih se zatim određuju osi niždimenzionalnog prostora. U slučaju nadziranog procesa, dokumente prvo dijelimo u skupine prema pripadnosti odgovarajućoj kategoriji. Stoga ćemo na početku imati onoliko novih osi koliko je i različitih kategorija. Postupak se dalje provodi tako da ove grupe dokumenata dijelimo na podgrupe, a zatim nove osi usmjerujemo prema njima. Bitno je zamijetiti da i nakon podjela grupa u podgrupe, svaka novostvorena grupa sadrži samo dokumente jedne kategorije (Karypis,2000b).

3.4 Pridruživanje težina izrazima

Pridruživanje težina izrazima možemo promatrati kao blagi oblik selekcije atributa, jer dok selekcija atributa u potpunosti uklanja neke atribute, pridruživanje težina mijenja samo njihov utjecaj (Joachims,2001).

Pridruživanje težina izrazima u većini se slučajeva bazira na tri standardne pretpostavke[18]:

1. rijetki izrazi nisu manje važni od čestih (IDF pretpostavka)
2. višestruka pojavljivanja nisu manje važna od jednostrukih pojavljivanja (TF pretpostavka)
3. dugački dokumenti nisu važniji od kratkih (normalizacijska pretpostavka).

S obzirom na pretpostavke, i težine izraza se obično sastoje od 3 komponente: komponente dokumenta, komponente kolekcije te normalizacijske komponente.

Intuitivno pridajemo veću važnost izrazima koji se učestalije pojavljuju. Stoga se kao osnovna mjera i navodi frekvencija izraza $TF(w_i,d_j)$. Frekvenciju izraza definiramo kao broj pojavljivanja riječi w_i u dokumentu d_j. Ona je osnovni gradivni element komponente dokumenta čiji su uobičajeni oblici navedeni u tablici 3.1.

komponenta dokumenta		
b	1.0	za izraze koji su prisutni u dokumentu uteg ima iznos 1, a inače 0
t	$TF(w_i,d)$	frekvencija izraza, broj ponavljanja izraza w_i u dokumentu d
n	$0.5 + 0.5\frac{TF(w_i,d)}{\max_j TF(w_j,d)}$	normalizirana frekvencija izraza (smanjena je razlika između čestih i rijetkih izraza)

Tablica 3.1 Uobičajene vrijednosti za komponentu dokumenta, preuzeto iz (Joachims,2001)

[18] J. Zobel, A. Moffat, (1998), Exploring the similarity space. SIGIR Forum, 32(1) pp.18-34, via (Debole,2003).

Sami podaci o frekvenciji izraza ne mogu osigurati dobre rezultate pri dohvatu informacija, pogotovo ako ti učestali izrazi nisu koncentrirani na dokumente određene kategorije, već su raspršeni po cijelom skupu. Kako bi takvim izrazima mogli pridružiti manju težinu, uvodimo komponentu kolekcije koja se bazira na vrijednosti frekvencije dokumenata DF(w_i). Nju definiramo kao broj dokumenata u kojima se pojavljuje izraz w_i. Sada je već jasno da nam najviše odgovaraju izrazi koji imaju veliku frekvenciju izraza, a malu frekvenciju dokumenata. Uobičajene vrijednosti komponente kolekcije navedene su u tablici 3.2.

komponenta kolekcije		
x	1.0	ignoriramo frekvenciju dokumenata
t	$\log \frac{\lvert D \rvert}{DF(w_i)}$	inverzna frekvencija dokumenata (idf), |D| je ukupni broj dokumenata u kolekciji. (izrazi koji se pojavljuju u više dokumenata će dobiti niže težine)
n	$\log \frac{\lvert D \rvert - DF(w_i)}{DF(w_i)}$	probabilistički inverz frekvencije dokumenata (izrazi koji se pojavljuju u više dokumenata će dobiti niže težine)

Tablica 3.2 Uobičajene vrijednosti za komponentu kolekcije, preuzeto iz (Joachims,2001)

Kod skupova sa većim varijacijama u duljinama dokumenata bitna je i treća, normalizacijska komponenta.

normalizacijska komponenta		
x	1.0	bez normalizacije
c	$\frac{1}{\sqrt{\sum x_j^2}}$	normalizacija na vektor duljine 1 u 2-normi
a	$\frac{1}{\sum x_j}$	normalizacija na vektor duljine 1 u 1-normi

Tablica 3.3 Uobičajene vrijednosti za normalizacijsku komponentu, preuzeto iz (Joachims,2001)

Koristimo li velik broj izraza za reprezentaciju dokumenata, očito je da će vektori duljih dokumenata biti popunjeniji i time imati i veću šansu da usporedbom s

izrazima iz upita budu označeni kao relevantni. Međutim, svi relevantni dokumenti trebali bi biti tretirani jednako bez obzira na duljinu, stoga je zadaća normalizacijske komponente da izjednači duljine vektora dokumenata (Salton,1998).

U tablicama 3.1-3.3 navedeni su najčešći oblici pojedinih komponenti. Preporučena kombinacija prema (Salton,1998) je *tfc*[19].

U (Debole,2003) izložen je i korak dalje pa se vrši nadzirano pridruživanje težina. Umjesto da se gleda raspodjela izraza po cijeloj kolekciji gledaju se razlike u raspodjelama kod pozitivnih i negativnih primjera iz kolekcije. Iako se govori o povećanju efikasnosti, ne može se govoriti o konstantnoj superiornosti nad tfc metodom.

[19] tfc kombinacija je u literaturi poznata i pod nazivom TFIDF.

4 Mjere uspješnosti

Kada govorimo o problemu klasifikacije teksta, čest je slučaj da se radi s malim brojem pozitivnih primjera. Ponekad tek oko 1% primjera iz skupa pripada traženoj kategoriji. Klasifikator, koji u tom slučaju sve primjere klasificira kao negativne, imat će točnost od 99%. Nažalost, takav nam klasifikator neće dohvatiti niti jedan pozitivan primjer. Time postaje potpuno bezvrijedan unatoč naoko impresivnim rezultatima. Nekad nam je bitna ravnoteža između netočno klasificiranih pozitivnih primjera i netočno klasificiranih negativnih primjera; nekad ćemo radije prihvatiti i veći broj netočno-pozitivnih primjera[20] ako će to značiti smanjenje broja netočno-negativnih primjera[21]. U tablici 4.1 definirana je notacija za sve moguće ishode. Definicije mjera uspješnosti preuzete su iz (Joachims, 2001) i (Sebastianini, 2002).

	klasa: y = +1	klasa: y = -1
predviđanje: $h(\vec{x}) = +1$	**točno-pozitivno TP**	**netočno-pozitivno NP**
predviđanje: $h(\vec{x}) = -1$	**netočno-negativno NN**	**točno-negativno TN**
$\sum$	ukupno pozitivnih	ukupno negativnih

Tablica 4.1 Mogući ishodi prilikom predviđanja klase

4.1 Točnost i pogreška

Dvije vjerojatno najintuitivnije mjere. Mogu se izračunati direktno iz tablice mogućih ishoda. Točnost i pogreška definiraju se izrazima (4.1) i (4.2) respektivno.

$$tocnost(h) = \Pr(h(\vec{x}) = y \mid h) \tag{4.1}$$

$$pogreska(h) = \Pr(h(\vec{x}) \neq y \mid h) \tag{4.2}$$

$$tocnost(h) + pogreska(h) = 1 \tag{4.3}$$

[20] Negativni primjeri pogrešno označeni kao pozitivni

[21] Pozitivni primjeri pogrešno označeni kao negativni

Procjenjuju se na temelju izraza (4.4) i (4.5).

$$tocnost_{test}(h) = \frac{TP + TN}{TP + NP + NN + TN} \tag{4.4}$$

$$pogreska_{test}(h) = \frac{NP + NN}{TP + NP + NN + TN} \tag{4.5}$$

Ove dvije mjere nisu osobito prikladne za vrednovanje rada klasifikatora teksta, jer pridaju jednaku važnost netočno-pozitivnim i netočno-negativnim ishodima. Što se tiče procesa klasifikacije teksta, veći se naglasak stavlja na točno predviđanje pozitivnih nego negativnih primjera.

Jedno od mogućih rješenja je pridavanje težina mogućim ishodima čime naglašavamo koji su nam ishodi od većeg, a koji od manjeg interesa.

4.2 Preciznost i odziv

Vjerojatnost da je dokument klase y = 1 točno kategoriziran nazivamo odziv. Definicija odziva klasifikacijskog pravila h dana je izrazom (4.6), a formula za izračunavanje izrazom (4.7).

$$odziv(h) = \Pr(h(\vec{x}) = 1 \mid y = 1, h) \tag{4.6}$$

$$odziv_{test}(h) = \frac{TP}{TP + NN} \tag{4.7}$$

Preciznost klasifikacijskog pravila h je vjerojatnost da je dokument označen kao $h(\vec{x}) = +1$, doista i točan. Definicija preciznosti dana je izrazom (4.8), a formula za izračunavanje izrazom (4.9).

$$preciznost(h) = \Pr(y = 1 \mid h(\vec{x}) = 1, h) \tag{4.8}$$

$$preciznost_{test}(h) = \frac{TP}{TP + NP} \tag{4.9}$$

4.3 Kombinacija mjera

Ni preciznost ni odziv nemaju smisla ako su odvojeni jedan od drugog. (Sebastianini,2002). Znamo da ukoliko označimo sve primjere kao pozitivne, imat ćemo savršen odziv, što bi nas moglo navesti na zaključak da se radi o efikasnom klasifikatoru, iako je to daleko od istine. Iz prakse je poznato kako se veće vrijednosti za odziv dobivaju na račun smanjenja vrijednosti za preciznost i obratno. Stoga je jasno da klasifikator treba vrednovati mjerom koja kombinira i preciznost i odziv. Postoje različite metode koje to i čine, a ovdje ćemo navesti tri:

1. prosječna preciznost jedanaest točki
2. točka izjednačenja
3. F_β mjera

4.3.1 Prosječna preciznost jedanaest točki

Rezultat dobivamo tako da, promjenama parametara, klasifikator dovodimo u situacije s odzivom 0.0, 0.1, 0.2, ..., 0.9, 1.0. Za tih jedanaest točaka izračunamo i pripadne preciznosti te na temelju njih izračunamo prosječnu preciznost. Ta prosječna preciznost je rezultat koji nam govori o kvaliteti našeg klasifikatora.

4.3.2 Točka izjednačenja

Točka izjednačenja je stanje klasifikatora kada odziv i preciznost imaju jednak iznos. Rezultat dobivamo na sličan način kao i za metodu prosječne preciznosti jedanaest točaka; varirajući parametre klasifikatora iscrtavamo ovisnost preciznosti o odzivu, točka izjednačenja je ona točka gdje se ova krivulja siječe s pravcem preciznost = odziv. Ova metoda počiva na pretpostavci da će se, dok monotono podižemo odziv od nule prema jedinici, monotono spuštati preciznost od vrijednosti blizu 1 prema nižim. Kada se ove dvije vrijednost izjednače dobili smo točku izjednačenja. Ukoliko ne postoji stanje kada su ove dvije vrijednosti jednake, uzima se aritmetička sredina najbližih točki.

4.3.3 F_β mjera

F_β mjera je harmonijska sredina odziva i preciznosti s težinom β kao parametrom koji, ovisno o iznosu, naglašava važnost ili odziva ili preciznosti. F_β mjera je definirana izrazom (4.10), a aproksimira se izrazom (4.11).

$$F_\beta(h) = \frac{(1+\beta^2)\cdot preciznost(h)\cdot odziv(h)}{\beta^2 \cdot preciznost(h) + odziv(h)} \tag{4.10}$$

$$F_{\beta,test}(h) = \frac{(1+\beta^2)\cdot TP}{(1+\beta^2)\cdot TP + NP + \beta^2 \cdot NN} \tag{4.11}$$

Iz formula je očito kako β utječe na odnos preciznosti i odziva unutar formule. Ako je $\beta = 0$, tada se ova mjera poistovjećuje s preciznošću, ako je $\beta = 1$[22] tada i preciznost i odziv imaju jednak utjecaj, no ako β ode u beskonačnost, tada ovu mjeru možemo poistovjetiti s odzivom (Sebastianini,2002).

4.4 Mikro i makro usrednjavanje

Ako želimo izračunati uspješnost klasifikatora na više trening ili test skupova, ili na više različitih kategorija, moramo pronaći način da obuhvatimo sve dobivene rezultate. Dva su osnovna načina za postizanje ovoga cilja: makro-usrednjavanje i mikro-usrednjavanje.

Makro-usrednjavanje odgovara standardnom pojmu aritmetičke sredine. Mjera efikasnosti računa se posebno za svaki od *m* eksperimenata, a zatim se izračuna aritmetička sredina. Primjer za F_1 mjeru dan je izrazom (4.12).

$$F_1^{macro} = \frac{1}{m}\sum_{i=1}^{m} F_1(h_1) \tag{4.12}$$

Mikro-usrednjavanje ne usrednjava rezultate eksperimenata (mjere efikasnosti), već usrednjava polja tablica mogućih ishoda. Izračunava se aritmetička sredina elemenata tablica i dobivaju vrijednosti: TP^{avg}, NP^{avg}, NN^{avg} i TN^{avg}. Mikrousrednjena F_1 mjera računa se prema izrazu (4.13).

$$F_1^{micro} = \frac{2\cdot TP^{avg}}{2\cdot TP^{avg} + NP^{avg} + NN^{avg}} \tag{4.13}$$

[22] Ovo je standardni izbor za parametar β.

5 Jezgrene metode

Kod nadziranog učenja, sustavu za učenje se predočuje skup primjera zajedno sa željenom klasifikacijom. Postoji veliki broj hipoteza koje bi mogle korektno izvršiti klasifikaciju. Među njima, linearne funkcije najbolje su proučene, a i najlakše se implementiraju.

Ovakvi linearni sustavi mogu se transformirati u oblik nazvan dualna reprezentacija, koji omogućava da se primjenom jezgrenih funkcija podaci preslikaju u visokodimenzionalni prostor kako bi se povećala ekspresivna moć linearnih klasifikatora.

5.1 Linearni klasifikator

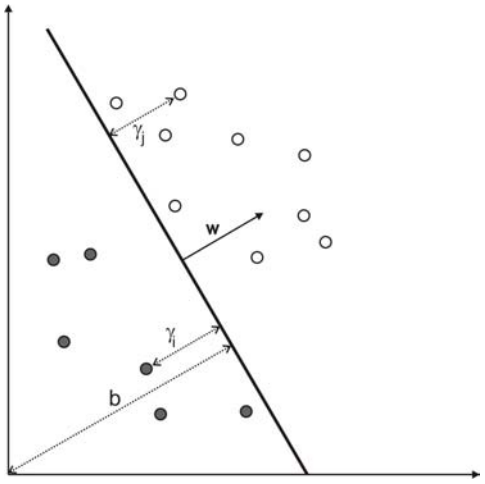
Binarnu klasifikaciju obavljamo koristeći se realnom funkcijom $f : X \subseteq R^n \to R$ na način da ulazni vektor $\mathbf{x} = (x_1,...,x_n)'$ klasificiramo kao pozitivan ukoliko vrijedi $f(\mathbf{x}) \geq 0$, a inače kao negativan. Ako se radi o linearnoj funkciju, može se zapisati u obliku izraza (5.1).

$$f(x) = \langle \mathbf{w} | \mathbf{x} \rangle + b \tag{5.1}$$

Parametre **w** i b određuje algoritam učenja na temelju primjera za učenje.

Geometrijski gledano, ulazni prostor je podijeljen hiperravninom[23], koja se definira izrazom (5.1), na dva dijela.

[23] Hiperravnina je podprostor dimenzije n-1 koji prostor dimenzije n dijeli na dva dijela (Cristianini, 2000).



Slika 5.1 Linearni klasifikator, razdvajajuća hiperravnina

Na slici 5.1 dan je geometrijski prikaz. Hiperravnina, koja je prikazana debljom linijom, određena je parametrima **w** i b. Parametar **w**, vektor težina, vektor je koji određuje smjer[24] ove hiperravnine, dok parametar b, odmak, određuje udaljenost hiperravnine od središta koordinatnog sustava. Očito je da je za prikaz bilo koje hiperravnine nužno i dovoljno odrediti n+1 parametar.

U problematici klasifikacije teksta, česte su ove oznake (Cristianini, 2000): $X \subseteq \mathbf{R}^n$ označava ulazni prostor, $\mathbf{Y} = \{-1,1\}$[25] označava izlaznu domenu. U skladu s ovim oznakama, skup primjera za učenje prikazuje se izrazom (5.2),

$$S = ((\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l)) \subseteq (X \times Y)^l \tag{5.2}$$

gdje *l* označava broj primjera za učenje.

Izrazom (5.3) definirat ćemo funkcijsku marginu[26] primjera ($\mathbf{x}_i$,y_i) u odnosu na hiperravninu (**w**,b).

$$\gamma_i = y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) \tag{5.3}$$

[24] Vektor **w** je okomit na hiperravninu.

[25] Ovdje smo naveli izlaznu domenu samo za slučaj binarne klasifikacije; razlog je taj što metoda koju ćemo prikazati radi isključivo binarnu klasifikaciju.

[26] Funkcijskom marginom primjera za učenje nazivamo rezultat izraza (5.3) za taj primjer za učenje, geometrijska margina primjera za učenje je rezultat izraza (5.3) ukoliko normiramo vektor **w** na jediničnu vrijednost..

Bitno je zamijetiti kako vrijednost $\gamma_i > 0$ implicira ispravnu klasifikaciju primjera ($\mathbf{x_i}$,y_i). Geometrijska interpretacija (slika 5.1) prikazuje kako margina primjera ($\mathbf{x_i}$,y_i) predstavlja euklidsku udaljenost[27] točke od hiperravnine.

Algoritam učenja linearnog klasifikatora kroz niz iteracija vrši korekcije vrijednosti vektora težine **w** i odmaka b sve dok ne bude zadovoljen neki od kriterija zaustavljanja[28].

[27] Riječ je o euklidskoj udaljenosti ukoliko je vektor težina **w** normiran.

[28] Prvi kriterij bio bi svakako korektna klasifikacija svih primjera, zatim maksimalan broj iteracija, spuštanje ispod maksimalne dopuštene pogreške, ...

5.2 Metoda potpornih vektora[29] - SVM

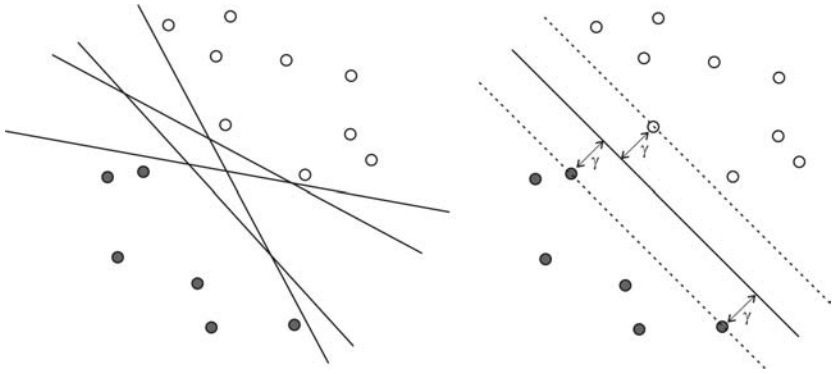
Metoda potpornih vektora bazira se na principu strukturne minimizacije rizika[30] koji pronalazi hipotezu h za koju se može garantirati najmanja vjerojatnost pogreške na skupu za učenje definiranom sa (5.2). Pokazano je kako se granica pogreška minimizira maksimiziranjem margine γ[31].

5.2.1 Klasifikator s maksimalnom[32] marginom

Pretpostavimo da je skup za učenje linearno razdvojiv, tj. da egzistira hiperravnina (**w**,b) tako da izraz (5.4) vrijedi za sve primjere skupa.

$$y_i\left(\langle \mathbf{w} \,|\, \mathbf{x}_i \rangle + b\right) > 0 \tag{5.4}$$

Tada postoji više hiperravnina koje razdvajaju skup za učenje bez pogreške.



Slika 5.2 Problem binarne klasifikacije u dvije dimenzije (Joachims, 2001)

---

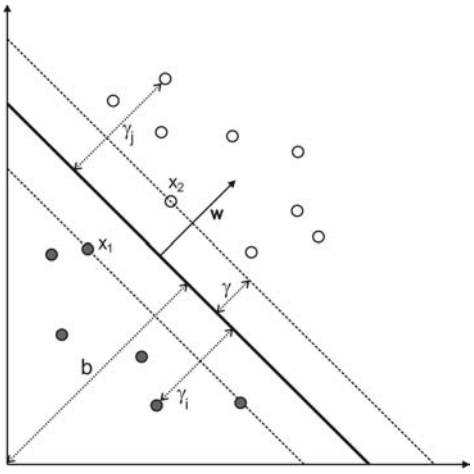
[29] eng. support vector machines

[30] eng. Structural Risk Minimization principle. Ovaj princip uveo je Vapnik (Vapnik,1982) a temelji se na statističkoj teoriji učenja.

[31] Funkcijska margina skupa za učenje je rezultat izraza (xii.3) za neki od potpornih vektora. Geometrijska margina skupa za učenje je rezultat izraza (xii.3) ukoliko umjesto **w** uvrstimo **w**/||**w**||, tj. normiramo vektor **w**.

[32] eng. maximal margin classifier. U literaturi se često ovaj klasifikator naziva i klasifikatorom s čvrstom marginom (eng. hard margin classifier) iz tog razloga što ne dopušta krivu klasifikaciju niti jednog primjera iz skupa za učenje.

Na slici 5.2 (lijevo) prikazane su neke od tih hiperravnina. Na istoj slici prikazana je i hiperravnina (desno) koju pronalazi metoda potpornih vektora, a karakterizira je maksimalna margina γ. Primjeri koji su najbliži margini, nazivaju se potpornim vektorima[33].



Slika 5.3 Klasifikator s maksimalnom marginom

Treba zamijetiti da se funkcija vezana uz hiperravninu (**w**,b) ne mijenja ako skaliramo hiperravninu na (λ**w**, λb), gdje je $\lambda \in \mathbf{R}^+$ (Cristianini,2000). Stoga možemo skalirati parametre hiperravnine tako da funkcijska margina iznosi 1. Dakle, za primjer sa slike 5.3, vrijedi izraz (5.5) iz čega se lako izvodi izraz za geometrijsku marginu (5.7) (Schölkopf,1997).

$$\begin{aligned} \langle \mathbf{w} | \mathbf{x}_2 \rangle + b &= +1 \\ \langle \mathbf{w} | \mathbf{x}_1 \rangle + b &= -1 \end{aligned} \tag{5.5}$$

$$\langle \mathbf{w} | \mathbf{x}_2 - \mathbf{x}_1 \rangle = 2 \tag{5.6}$$

$$\begin{aligned} \gamma &= \frac{1}{2} \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|} \middle| \mathbf{x}_2 - \mathbf{x}_1 \right\rangle \\ \gamma &= \frac{1}{\|\mathbf{w}\|} \end{aligned} \tag{5.7}$$

Lako je uočiti kako je minimalna udaljenost između bilo koja dva primjera različitih klasa jednaka dvostrukoj margini definiranoj s (5.7). Kako bi maksimizirali marginu trebamo pronaći:

[33] eng. support vectors

$$\min\left[\frac{1}{2}\left\|\mathbf{w}^2\right\|\right] \tag{5.8}$$

uz zadovoljene uvjete:

$$y_i\left(\left\langle \mathbf{w}\middle|\mathbf{x}_i\right\rangle + b\right) \geq 1, \quad i = 1 \ldots l \tag{5.9}$$

Numerički je ovaj problem veoma teško riješiti, iako se radi o konveksnom problemu, stoga ga nastojimo transformirati u dualnu formu. Izrazom (5.10) dan je Lagrangian ovog problema.

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\left\langle \mathbf{w}\middle|\mathbf{w}\right\rangle - \sum_{i=1}^{l} \alpha_i [y_i(\left\langle \mathbf{w}\middle|\mathbf{x}_i\right\rangle + b) - 1] \tag{5.10}$$

Gdje su $\alpha_i \geq 0$ Lagrangeovi množitelji[34]. Uvjet da u sedlu, derivacije od L po primalnim varijablama moraju nestati, vodi do izraza (5.11).

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = \mathbf{0} \\ \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} &= \sum_{i=1}^{l} y_i \alpha_i = 0 \end{aligned} \tag{5.11}$$

Transformacijama ovih izraza dobivamo:

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i \\ 0 &= \sum_{i=1}^{l} y_i \alpha_i \end{aligned} \tag{5.12}$$

što nas uvrštavanje u Lagrangian vodi do dualne forme koju je potrebno maksimizirati.

$$\begin{aligned} L(\mathbf{w}, b, \alpha) &= \frac{1}{2}\left\langle \mathbf{w}\middle|\mathbf{w}\right\rangle - \sum_{i=1}^{l} \alpha_i [y_i(\left\langle \mathbf{w}\middle|\mathbf{x}_i\right\rangle + b) - 1] \\ &= \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle x_i \middle| x_j \right\rangle - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle x_i \middle| x_j \right\rangle + \sum_{i=1}^{l} \alpha_i \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle x_i \middle| x_j \right\rangle \end{aligned} \tag{5.13}$$

Dakle, imamo zadan skup za učenje izrazom (5.2), neka je α^* rješenje dualne forme, izraz (5.14) uz uvjete iz (5.15), kvadratnog optimizacijskog problema originalno zadanog izrazom (5.8) uz uvjete iz (5.9).

---

[34] Optimizacijski postupci, a među njima i teorija Lagrangiana opisani su u (Cristianini,2000) pp.79-92

$$\max\left[W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle x_i \middle| x_j \right\rangle \right] \tag{5.14}$$

$$\begin{aligned} &\sum_{i=1}^{l} y_i \alpha_i = 0 \\ &\alpha_i \geq 0, \quad i = 1 \ldots l \end{aligned} \tag{5.15}$$

Tada vektor težina $\mathbf{w}^*$, izraz (5.16), daje hiperravninu s maksimalnom marginom čija je geometrijska margina dana izrazom (5.17).

$$\mathbf{w}^* = \sum_{i=1}^{l} y_i \alpha_i^* \mathbf{x}_i \tag{5.16}$$

$$\gamma = \frac{1}{\left\| \mathbf{w}^* \right\|} \tag{5.17}$$

Kako se odmak b ne pojavljuje nigdje u dualnoj formulaciji problema, mora ga se izračunati iz primalnih varijabli, izraz (5.18).

$$b^* = -\frac{\max_{y_i=-1}(\left\langle \mathbf{w}^* \middle| \mathbf{x}_i \right\rangle) + \min_{y_i=+1}(\left\langle \mathbf{w}^* \middle| \mathbf{x}_i \right\rangle)}{2} \tag{5.18}$$

Prema Kuhn-Tuckerovom teoremu, postoji i dopunski uvjet[35], uz one iz (5.15), dan izrazom (5.19) koji ukazuje na strukturu rješenja.

$$\alpha_i [y_i (\left\langle \mathbf{x}_i \middle| \mathbf{w} \right\rangle + b) - 1] = 0, \quad i = 1 \ldots l \tag{5.19}$$

Lako se da zaključiti kako su jedino za one primjere $\mathbf{x_i}$, za koje funkcijska margina iznosi 1, te leže točno na geometrijskoj margini odgovarajući α_i^* različiti od nule. Ostali su primjeri nevažni; automatski zadovoljavaju uvjete iz (5.15) i ne sudjeluju u konstrukciji vektora težina $\mathbf{w}^*$, izraz (5.16)[36].

Kada bi izostavili neki primjer iz skupa za učenje, označimo ga s $\mathbf{x_i}^*$, konstruirali rješenje i zatim provjerili kako naš sustav klasificira izostavljeni primjer, našli bi se u jednoj od četiri situacije (Schölkopf,1997):

1. $y_{i*} \cdot (\left\langle \mathbf{x}_{i*} \middle| \mathbf{w} \right\rangle + b) > 1$ - Primjer je dobro klasificiran i ne leži na margini; stoga ne bi postao potporni vektor pa ne bi niti utjecao na krajnji rezultat.

[35] Radi se o uvjetu poznatijem pod nazivom Karush-Kuhn-Tuckerov dopunski uvjet. Detalji oko Kuhn-Tuckerova teorema mogu se pogledati u (Cristianini,2000) pp.87.

[36] Primjeri za koje je α_i^* različit od nule, nazivaju se potpornim vektorima iz razloga što sudjeluju u formiranju vektora težina $\mathbf{w}^*$.

2. $y_{i*} \cdot (\langle \mathbf{x}_{i*} | \mathbf{w} \rangle + b) = 1$ - Primjer je dobro klasificiran, leži na margini, postao bi potporni vektor, ali vektor težina **w** se ne bi promijenio.
3. $1 > y_{i*} \cdot (\langle \mathbf{x}_{i*} | \mathbf{w} \rangle + b) > 0$ - Primjer je dobro klasificiran, leži unutar margine, ali s prave strane hiperravnine. Postao bi potporni vektor i promijenio bi rješenje **w**.
4. $y_{i*} \cdot (\langle \mathbf{x}_{i*} | \mathbf{w} \rangle + b) < 0$ - Primjer nije dobro klasificiran, leži s krive strane hiperravnine. Postao bi potporni vektor i promijenio bi rješenje **w**.

Razdvajajuća hiperravnina se također može prikazati u dualnoj formi, izraz (5.20).

$$\begin{aligned} f(\mathbf{x}, \alpha^*, b^*) &= \sum_{i=1}^{l} y_i \alpha_i^* \langle \mathbf{x}_i | \mathbf{x} \rangle + b^* \\ &= \sum_{i \in sv}^{l} y_i \alpha_i^* \langle \mathbf{x}_i | \mathbf{x} \rangle + b^* \end{aligned} \tag{5.20}$$

Oznaka sv predstavlja skup potpornih vektora. Vidljivo je da Lagrangeovi množitelji određuju utjecaj pojedinog primjera iz skupa za učenje na rješenje.

5.2.2 Metoda potpornih vektora sa slabom marginom[37]

Nedostatak prethodne formulacije problema je taj što učenje neće biti uspješno ukoliko primjeri nisu linearno razdvojivi[38]. Ukoliko skup za učenje sadrži šum, što je karakteristika većine realnih skupova, u općenitom slučaju kategorije neće biti linearno razdvojive. Stoga je potrebno razviti sofisticiraniju metodu kojom će se ovaj problem moći riješiti.

Uvođenjem slabe margine, dopušta se da određeni primjeri budu pogrešno naučeni. Kako bi se modelirala slaba margina, mijenjaju se uvjeti iz (5.9) u (5.21) uvođenjem varijabli ξ_i koje dopuštaju primjerima ne samo da stoje izvan granica margine već i da budu krivo klasificirani.

$$\begin{aligned} & y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad i = 1 \ldots l \\ & \xi_i \geq 0, \quad i = 1 \ldots l \end{aligned} \tag{5.21}$$

Međutim, uvjeti iz (5.21) dopuštaju učenje s po volji velikom pogreškom. Da se ta pogreška učini što manjom, treba modificirati i izraz (5.8). U praksi se uglavnom koriste sljedeće dvije metode: slaba margina u L_2 normi (izraz (5.22) uz uvjete iz (xi.23))

$$\min\left[\frac{1}{2}\|\mathbf{w}^2\| + \frac{1}{2}C\sum_{i=1}^{l}\xi_i^2\right] \tag{5.22}$$

$$y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad i = 1 \ldots l \tag{5.23}$$

te slaba margina u L_1 normi (izraz (5.24) uz uvjete iz (5.25)).

$$\min\left[\frac{1}{2}\|\mathbf{w}^2\| + \frac{1}{2}C\sum_{i=1}^{l}\xi_i\right] \tag{5.24}$$

---

[37] eng. soft margin classifier

[38] Iako se smatra kako je većina problema klasifikacije teksta linearno razdvojiva, ipak se nastoji dopustiti i određena odstupanja u skupu za učenje (Cristianini,2000). Glavni problem klasifikatora s maksimalnom marginom je taj što za rezultat uvijek daje hipotezu bez pogreške na skupu za učenje. Učenje će biti uspješno, iako podaci možda sadrže i šum, preslikamo li podatke u visokodimenzionalni prostor, koristeći se nekom od složenijih jezgrenih funkcija (pojam jezgrenih funkcija objašnjen je u poglavlju 5.3). Međutim, to bi nas dovelo do prenaučenosti.

$$
\begin{aligned}
& y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad i = 1 \ldots l \\
& \xi_i \geq 0, \quad i = 1 \ldots l
\end{aligned}
\tag{5.25}
$$

Optimalna vrijednost parametra C je apriori nepoznata te se određuje krosvalidacijom na skupu za učenje. Vidljivo je da je izbačen uvjet o pozitivnosti parametara ξ_i, izraz (5.23), ali to je učinjeno zbog toga što nam kvadratna forma u izrazu (5.22) dopušta i negativne vrijednosti parametara ξ_i.

Lagrangiani su dani izrazima (5.26) i (5.27).

$$
\begin{aligned}
& L(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}\langle \mathbf{w} | \mathbf{w} \rangle + \frac{C}{2}\sum_{i=1}^{l} \xi_i^2 - \sum_{i=1}^{l} \alpha_i [y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) - 1 + \xi_i] \\
& \alpha_i \geq 0, \quad i = 1 \ldots l
\end{aligned}
\tag{5.26}
$$

$$
\begin{aligned}
& L(\mathbf{w}, b, \alpha, \xi, \mathbf{r}) = \frac{1}{2}\langle \mathbf{w} | \mathbf{w} \rangle + C\sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i [y_i(\langle \mathbf{w} | \mathbf{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{l} r_i \xi_i \\
& \alpha_i \geq 0, \quad i = 1 \ldots l \\
& r_i \geq 0, \quad i = 1 \ldots l
\end{aligned}
\tag{5.27}
$$

Nakon nekoliko koraka ekvivalentnih onima za klasifikator sa čvrstom, tj. maksimalnom marginom dobivamo dualne definicije problema, izrazi (5.28)[39] i (5.29).

$$
\begin{aligned}
& \max\left[W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j (\langle x_i | x_j \rangle + \frac{1}{C}\delta_{ij}) \right] \\
& \sum_{i=1}^{l} y_i \alpha_i = 0 \\
& \alpha_i \geq 0, \quad i = 1 \ldots l
\end{aligned}
\tag{5.28}
$$

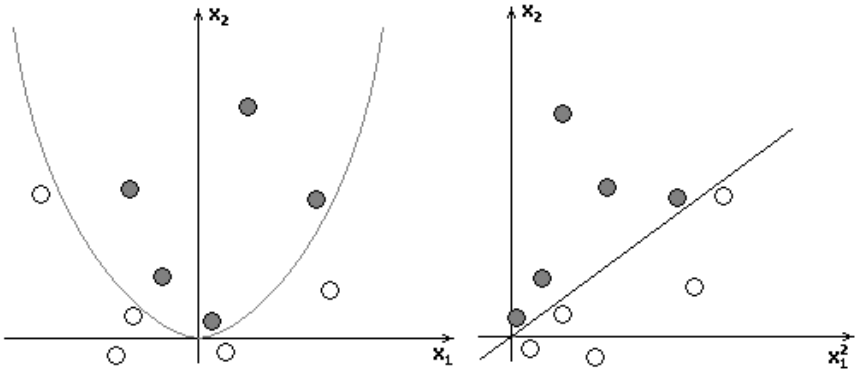
$$
\begin{aligned}
& \max\left[W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i | x_j \rangle \right] \\
& \sum_{i=1}^{l} y_i \alpha_i = 0 \\
& C \geq \alpha_i \geq 0, \quad i = 1 \ldots l
\end{aligned}
\tag{5.29}
$$

[39] U izrazu (5.28) pojavljuje se simbol δ_{ij}, poznatiji kao Kroneckerov δ, čija je vrijednost 1 ukoliko je i=j, a inače mu je vrijednost 0.

5.3 Jezgrene funkcije

Za većinu stvarnih aplikacija mogućnosti linearne funkcije nisu dovoljno ekspresivne, odnosno željeni koncept često se ne može opisati linearnom kombinacijom danih atributa. Jedan od načina na koji se ovaj problem može riješiti je izgradnja nelinearnog klasifikatora čiji je prostor hipoteza dovoljno ekspresivan da može naučiti željene koncepte. Međutim, promjena domene rada vjerojatno bi uzrokovale i potrebu za ponovnom promjenom klasifikatora. Primjer su neuronske mreže koje vrlo dobro pokazuju ovisnost građe klasifikatora o problematici.

Alternativan način rješavanja ovoga problema je pomoću jezgrenih metoda koje ne mijenjaju klasifikator, već preslikavaju podatke iz prostora gdje nisu linearno razdvojivi u prostor gdje to jesu.



Slika 5.4 Primjer preslikavanja u prostor karakteristika (Joachims,2001)

Primjer imamo na slici 5.4. S lijeve strane slike graf pokazuje skup podataka koji nije linearno razdvojiv u (x_1, x_2), dok se s desne strane nalazi prikazan ovaj isti problem, ali preslikan u (x_1^2, x_2) gdje je linearno razdvojiv.

Iako smo već predstavili koncept metode potpornih vektora, jedan njen izuzetno važan dio još nismo spomenuli, a to je činjenica kako se ova metoda vrlo lako transformira u nelinearni klasifikator.

Primjeri za učenje preslikavaju se u visokodimenzionalni prostor karakteristika nekom nelinearnom funkcijom ϕ[40]. Metoda potpornih vektora nauči linearnu hipotezu, koja korektno klasificira primjere za učenje u prostoru karakteristika. Iako je hipoteza linearna u prostoru karakteristika, preslikamo li je natrag, u prostor atributa, dobit ćemo nelinearnu hipotezu.

Kako bi naučili nelinearne relacije linearnim klasifikatorom, prvo moramo odabrati skup nelinearnih karakteristika koji će činiti prostor karakteristika, te preslikati ulazne podatke u taj prostor. Prostor hipoteza koji nam je sada na raspolaganju, a ekvivalentan je onome iz (5.1) za učenje nad prostorom atributa, dan je izrazom (5.30).

$$f(x) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}) + b \tag{5.30}$$

Nakon preslikavanja učimo klasifikator nad prostorom karakteristika.

Metoda potpornih vektora može se prikazati i u dualnoj formi, tj. i problem, izraz (5.31), i hipoteza, izraz (5.32), se mogu prikazati kao kombinacija primjera za učenje.

$$\begin{aligned} &\max\left[W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle \phi(\mathbf{x}_i) \middle| \phi(\mathbf{x}_j) \right\rangle \right] \\ &\sum_{i=1}^{l} y_i \alpha_i = 0 \\ &\alpha_i \geq 0, \quad i = 1 \ldots l \end{aligned} \tag{5.31}$$

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \left\langle \phi(\mathbf{x}_i) \middle| \phi(\mathbf{x}) \right\rangle + b \tag{5.32}$$

Uvodimo definiciju jezgrene funkcije (Cristianini,2000): jezgrena funkcija je funkcija K tako da za sve **x**,**z** iz X vrijedi:

$$K(\mathbf{x}, \mathbf{z}) = \left\langle \phi(\mathbf{x}) \middle| \phi(\mathbf{z}) \right\rangle \tag{5.33}$$

gdje je ϕ funkcija preslikavanja iz prostora atributa X u prostor karakteristika F.

Ako imamo zadan skup primjera za učenje s (5.2), tada su sve informacije potrebne jezgrenoj metodi u cijelosti sadržane u matrici skalarnih produkata

[40] $\phi : X \to F$ je nelinearno preslikavanje iz ulaznog prostora, koji se često naziva i prostor atributa, u prostor karakteristika.

prostora karakteristika, izraz (5.34) poznatoj kao Gramova matrica (Cristianini,2001).

$$G = K = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{l} \tag{5.34}$$

Traženje uvjetnog maksimuma izraza (5.31) te vrednovanje funkcije odlučivanja (5.32) zahtijevaju računanje skalarnog produkta $<\phi(\mathbf{x}_i)|\,\phi(\mathbf{x}_j)>$ u visokodimenzionalnom prostoru. Uz ispunjene uvjete iz Mercerovog teorema[41], ovaj skupi račun može se znatno reducirati korištenjem prikladne jezgrene funkcije (Schölkopf,1997). Rezultat su izrazi (5.34) te (5.35).

$$\begin{aligned} &\max\left[W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right] \\ &\sum_{i=1}^{l} y_i \alpha_i = 0 \\ &\alpha_i \geq 0, \quad i = 1 \ldots l \end{aligned} \tag{5.31}$$

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{5.32}$$

Funkcijska forma preslikavanja ϕ ne mora biti poznata jer je implicitno zadana izborom jezgrene funkcije (Campbell,2000).

5.3.1 Primjeri jezgrenih funkcija

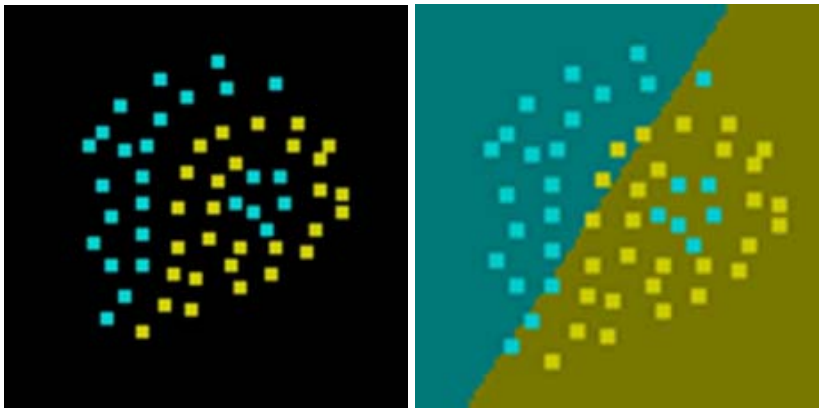
Primjere i svojstva nekih jezgrenih funkcija upoznat ćemo preko rezultata klasifikatora na skupu za učenje sa slike 5.5.

Jezgrena je funkcija skalarni produkt u višedimenzionalnom Hilbertovom prostoru (Campbell,2000). Ukoliko je taj ciljni prostor jednak početnom, imamo trivijalan slučaj i linearnu jezgrenu funkciju, izraz (5.33).

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} | \mathbf{z} \rangle \tag{5.33}$$

Uz linearnu jezgrenu funkciju imamo i linearni klasifikator. Mogući rezultat učenja za problem sa slike 5.5 (lijevo) dan je na slici 5.5 (desno).

[41] Mercerov teorem daje uvjete koji neku funkciju K(**x**,**z**) čine jezgrenom funkcijom. Iskaz teorema dan je u (Schölkopf,1997, pp.27-29) te u (Cristianini,2000, pp.33-38), uz znatnu teorijsku podršku te se ovdje neće posebno navoditi.



Slika 5.5 Skup primjera za učenje (lijevo), rezultati klasifikatora s linearnom jezgrenom funkcijom (desno)

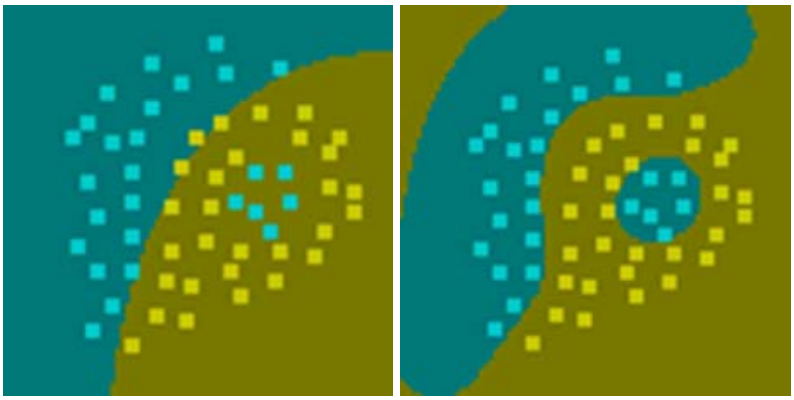
Polinomijalna jezgrena funkcija dana je izrazom (5.34), uz d ≥ 2.

$$K(\mathbf{x},\mathbf{z}) = (\langle \mathbf{x} | \mathbf{z} \rangle + c)^d \tag{5.34}$$

Prikaz rezultata učenja pomoću polinomijalne jezgrene funkcije stupnja 3 dan je na slici 5.6 (lijevo).

Izraz (5.35) predstavlja RBF[42] jezgrenu funkciju. Njeni rezultati dani su na slici 5.6 (desno).

$$K(\mathbf{x},\mathbf{z}) = \exp(-\gamma \|\mathbf{x}-\mathbf{z}\|^2) \tag{5.35}$$



Slika 5.6 Rezultati učenja uz pomoć polinomijalne jezgrene funkcije stupnja 3 (lijevo), a uz pomoć RBF jezgrene funkcije (desno)

[42] eng. radial basis function

Različiti izbori jezgrene funkcije definiraju različite prostore karakteristika te će i dobiveni klasifikatori dati različite rezultate. Utjecaj jezgrene funkcije na efikasnost klasifikatora lako se da uočiti iz primjera navedenog na slikama 5.5 i 5.6, uz napomenu kako jezgrenu funkciju treba odabrati u skladu s problematikom zadatka. Dva su pristupa odabiru, odnosno kreiranju jezgrene funkcije. Prvi je stvoriti prostor karakteristika koji najbolje odgovara podacima, a onda pronaći odgovarajuću jezgrenu funkciju koja razapinje taj prostor. Drugim se načinom, češće korištenim u praksi, odmah definira jezgrena funkcija čime implicitno definiramo prostor karakteristika i izbjegavamo bilo kakav rad u ili na njemu, te se zatim vrednuje naš izbor metodom poput krosvalidacije.

5.4 Jezgrene funkcije za klasifikaciju teksta

Većina metoda dokumente uspoređuje na razini izraza, odnosno dokumenti su slični koriste li iste izraze. Nedostatak ovakvih metoda leži u činjenici što ne obuhvaćaju semantičke veze među izrazima, već izraze promatraju kao izolirane pojave. Stoga su dokumenti koji govore o istoj temi, ali različitim izrazima preslikani u veoma udaljena područja prostora karakteristika. Dakle, tražimo preslikavanje koje bi bilo izvedeno semantičkom jezgrenom funkcijom koja bi određivala sličnost među dokumentima uzimajući u obzir i povezanost različitih izraza (Kandola,2002).

5.4.1 Osnovne jezgrene funkcije za klasifikaciju teksta

Krenimo od prikaza teksta. Neka je svaki dokument prikazan vektorom $\mathbf{d}_i$, izraz (5.36), čija je dimenzija jednaka broju različitih izraza u skupu primjera za učenje D, izraz (5.37).

$$\mathbf{d}_i = [t_{1i}\, t_{2i} \ldots t_{ni}]' \tag{5.36}$$

$$\begin{aligned} D &= [\mathbf{d}_1\, \mathbf{d}_2 \ldots \mathbf{d}_l] \\ &= \begin{bmatrix} t_{11} & \cdots & t_{1l} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nl} \end{bmatrix} \\ &= [\mathbf{t}_1\, \mathbf{t}_2 \ldots \mathbf{t}_n]' \end{aligned} \tag{5.37}$$

U izrazu (5.36), t_{ji} predstavlja broj pojavljivanja izraza *j* u dokumentu *i*. Na temelju matrice D (izraz-dokument matrica[43]) definiramo matricu G (dokument-dokument matrica[44]), izraz (5.38) te matricu T (izraz-izraz matrica[45]), izraz (5.39).

$$\begin{aligned} G &= D'D \\ &= \begin{bmatrix} \mathbf{d}_1'\mathbf{d}_1 & \cdots & \mathbf{d}_1'\mathbf{d}_l \\ \vdots & \ddots & \vdots \\ \mathbf{d}_l'\mathbf{d}_1 & \cdots & \mathbf{d}_l'\mathbf{d}_l \end{bmatrix} \end{aligned} \tag{5.38}$$

[43] eng. term by document matrix

[44] eng. document by document matrix

[45] eng. term by term matrix

$$T = DD' = \begin{bmatrix} \mathbf{t}_1\mathbf{t}_1' & \cdots & \mathbf{t}_1\mathbf{t}_n' \\ \vdots & \ddots & \vdots \\ \mathbf{t}_n\mathbf{t}_1' & \cdots & \mathbf{t}_n\mathbf{t}_n' \end{bmatrix} \quad (5.39)$$

Ako učenje obavljamo u originalnom ulaznom prostoru, prostor atributa, jezgrena nam je funkcija definirana s (5.40) i očito je kako Gramovu matricu zapravo predstavlja dokument-dokument matrica G.

$$K(\mathbf{d}_i, \mathbf{d}_j) = \left\langle \mathbf{d}_i \middle| \mathbf{d}_j \right\rangle = \mathbf{d}_i' \mathbf{d}_j \quad (5.40)$$

U općenitijem slučaju trebamo obuhvatiti i preslikavanja primjera (vektora dokumenata) u prostor karakteristika pomoću neke funkcije ϕ. Najjednostavniji slučaj je linearno preslikavanje, definirano izrazom (5.41), gdje je P neka matrica.

$$\phi(\mathbf{d}) = P\mathbf{d} \quad (5.41)$$

U ovom slučaju, jezgrena funkcija dana je izrazom (5.42), a pripadajuća Gramova matrica izrazom (5.43).

$$K(\mathbf{d}_i, \mathbf{d}_j) = \mathbf{d}_i' P' P \mathbf{d}_j \quad (5.42)$$

$$G = D' P' P D \quad (5.43)$$

Pridruživanje težinskih faktora izrazima, objašnjeno u poglavlju 3.4, izvodi se tako da se na dijagonalu matrice P postave težinski faktori za pojedine izraze. Vrijednost elementa na dijagonali, P(i,i), jednaka je težinskom faktoru za izraz *i*, dok su ostali elementi matrice P jednaki nuli.

Kao što je više puta naglašeno, ovakav prikaz modelira izraze kao međusobno nekorelirane, te im dodjeljuje ortogonalne smjerove u prostoru karakteristika. Što znači da se na temelju takvog prikaza mogu grupirati samo dokumenti koji imaju mnogo zajedničkih izraza (Cristianini,2001).

Jedno od rješenja je ekspanzija dokumenata tako da se dokumentu dodaju sinonimi i riječi bliske po značenju onim riječima koje dokument sadrži. Kako bi ovu metodu inkorporirali u jezgrenu funkciju, tražimo korelacije među izrazima. Izrazi su semantički povezani ako se često zajedno pojavljuju u istim dokumentima. Jezgra koja modelira ovaj princip dana je izrazom (5.44).

$$K(\mathbf{d}_i, \mathbf{d}_j) = (D'\mathbf{d}_i)'(D'\mathbf{d}_j) = \mathbf{d}_i' DD' \mathbf{d}_j \quad (5.44)$$

Matrica DD' je izraz-izraz matrica te je *ij* element ove matrice različit od nule, samo ako postoji dokument u kojem se izraz *i* i izraz *j* zajedno pojavljuju. Stoga, ova jezgrena funkcija uzima u obzir i semantičku povezanost izraza[46].

Informacije o povezanosti izraza mogu se dobiti i iz semantičke mreže[47]. Udaljenost među traženim izrazima u semantičkoj mreži obrnuto je proporcionalna njihovoj povezanosti. Ovakvo znanje može se uvrstiti u matricu P, na način da elementi $P_{ij} = P_{ji}$ izražavaju bliskost izraza *i* i *j*. Rezultat ovakvog nastojanja je jezgrena funkcija iz izraza (5.45).

$$K(\mathbf{d}_i, \mathbf{d}_j) = \mathbf{d}_i' P' P \mathbf{d}_j = \mathbf{d}_i' P^2 \mathbf{d}_j \tag{5.45}$$

5.4.2 Latentno semantičke jezgrene funkcije[48]

Metoda latentnog semantičkog indeksiranja LSI, opisana u poglavlju 3.3.4, koja obuhvaća semantičke informacije u mjeru sličnosti među dokumentima, u ovom će se poglavlju primjeniti za konstrukciju jezgrene funkcije.

SVD dekompozicija matrice D (izraz-dokument) dana je formulom (5.46) gdje je Σ dijagonalna[49] matrica, a matrice U i V su ortogonalne[50].

$$D = U \Sigma V' \tag{5.46}$$

Stupci matrice U su singularni vektori, poredani prema padajućim singularnim vrijednostima. Matricu P potrebnu za konstrukciju jezgrene funkcije, u ovom slučaju, definiramo izrazom (5.47), gdje je I_k jedinična matrica kojoj je samo prvih k dijagonalnih elemenata različito od nule, dok je U_k matrica koja se sastoji od prvih k stupaca matrice U.

$$P = U_k' = I_k U' \tag{5.47}$$

Kreirana jezgrena funkcija dana je izrazom (5.48).

---

[46] Semantička povezanost izraza obuhvaćena je pretpostavkom da su izrazi koji se često zajedno pojavljuju semantički povezani.

[47] Semantička mreža, za svaku riječ iz rječnika, definira hijerarhijsku povezanost s drugim riječima. Primjer su riječi "suprug" i "supruga", koje su specijalni slučajevi2 riječi "supružnik" (Cristianini,2001).

[48] eng. latent semantic kernels

[49] Svi neprazni elementi matrice se nalaze na dijagonali.

[50] U je ortogonalno matrica, ako zadovoljava uvjet UU'=U'U=I.

$$
\begin{aligned}
K(\mathbf{d}_i,\mathbf{d}_j) &= \mathbf{d}_i{}'P'P\mathbf{d}_j \\
&= \mathbf{d}_i{}'(I_kU')'(I_kU')\mathbf{d}_j \\
&= \mathbf{d}_i{}'UI_kU'\mathbf{d}_j
\end{aligned}
\tag{5.48}
$$

Smanjenjem dimenzionalnosti na k dimenzija novog prostora postigli smo efekt da su se dobro povezane dimenzije starog prostora[51] spojile u jednu dimenziju novog prostora. Na ovaj način možemo govoriti o obuhvaćanju semantičkog značenja zamjenom izraza matematički dobivenim konceptima. Definiramo li matricu P u kvadratno simetričnoj formi, izraz (5.49) dobit ćemo istu jezgrenu funkciju, (5.50).

$$
P = (U_kU')' = (UI_kU')' \tag{5.49}
$$

$$
\begin{aligned}
K(\mathbf{d}_1,\mathbf{d}_2) &= (UU_k{}'\mathbf{d}_1)'(UU_k{}'\mathbf{d}_2) \\
&= \mathbf{d}_1{}'U_kU'UU_k{}'\mathbf{d}_2 \\
&= \mathbf{d}_1{}'P'P\mathbf{d}_2
\end{aligned}
\tag{5.50}
$$

Matricu P možemo promatrati kao izraz-izraz matricu koja definira sličnost među izrazima. Zanimljivo je kako se jednako preslikavanje može izvršiti i implicitno radeći na manjoj dokument-dokument matrici (Cristianini,2001). Originalna izraz-dokument matrica D daje jezgrenu matricu:

$$
K = D'D \tag{5.51}
$$

Primjenom SVD dekompozicije dobivamo sljedeći rezultat:

$$
\begin{aligned}
K &= D'D \\
&= V\Sigma U'U\Sigma V' \\
&= V\Sigma^2V' \\
&= V\Lambda V'
\end{aligned}
\tag{5.52}
$$

Prostor karakteristika koji nastaje nakon odabira prvih k singularnih vrijednosti u potpunosti odgovara preslikavanju vektora karakteristika **d** u vektor $UI_kU'\mathbf{d}$. Rezultat je jezgrena matrica dana izrazom (5.53).

$$
\begin{aligned}
K &= D'UI_kU'UI_kU'D \\
&= V\Sigma U'UI_kU'U\Sigma V' \\
&= V\Sigma I_k\Sigma V' \\
&= V\Lambda_kV'
\end{aligned}
\tag{5.53}
$$

Nova jezgrena matrica može se izračunati direktno iz K obavljajući dekompoziciju po svojstvenim vrijednostima te množeći komponente nazad uz iznimku što se sve

[51] Riječ je o izrazima koji se često zajedno pojavljuju u dokumentima.

svojstvene vrijednosti osim prvih k postavljaju na nulu. Dakle, možemo imati jezgrenu funkciju koja odgovara LSI prostoru karakteristika bez da računamo te karakteristike (Cristianini,2001).

5.4.3 Ostale jezgrene funkcije

U (Lodhi,2002) dan je prikaz metode s jezgrenom funkcijom koja daje skalarni produkt između podnizova duljine k. Podnizom se smatra svako uređeno pojavljivanje niza znakova duljine k iako nije kontinuirano. Primjer je podniz "trk" koji se pojavljuje i u "**trk**ač" i u "**tr**a**k**tor". Dakle dokument se promatra kao dugi niz znakova.

U (Kandola,2002) opisana je metoda koja primjenjuje difuzijske jezgrene funkcije[52]. Ova metoda promatra matrice izraz-izraz i dokument-dokument kao usmjerene težinske grafove. Sličnost među izrazima definira preko strukture tih grafova, odnosno preko svih mogućih putova od jednog izraza do drugog kroz zadani graf.

U (Vinkourov,2002) opisana je metoda Jezgrene kanonske korelacijske analize[53] kojom se na temelju korpusa na dva različita jezika uspijeva dobiti prikaz teksta neovisan o jeziku. Korištenjem jezgrenih funkcija ova dva dijela korpusa se preslikaju u dva visokodimenzionalna prostora. Promatra se korelacija između ta dva prostora, te se odabiru dimenzije iz oba prostora koje su međusobno maksimalno korelirane. Kako se radi o dva različita jezika, pretpostavljamo kako su prikazi potpuno nezavisni osim semantičkog sadržaja u njima, stoga bi bilo kakva korelacija među njima trebala odražavati semantičku sličnost.

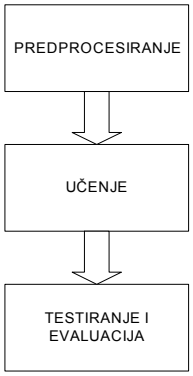
---

[52] eng. diffusion kernels, I. Kondor, (2001), *Diffusion kernels on continuous spaces* (Technical report), Carnegie Mellon University. via (Kandola,2002).

[53] eng. kernel canonical correlation analysis

6 Implementacija

Implementacija opisanih metoda može se podijeliti u nekoliko segmenata. Prvi bi bio predprocesiranje ulaznih podataka, drugi obrada, odnosno učenje nad tim podacima, a treći testiranje i vrednovanje rezultata.



Slika 6.1 Grubi prikaz implementacije

6.1 Ulazni podaci

Kao ulazni podaci na raspolaganju su dvije baze novinskih članaka Reuters-21578, te Vjesnik - baza članaka iz hrvatskih dnevnih novina Vjesnik iz razdoblja od 2000. do 2003. godine.

6.1.1 Baza članaka Reuters-21578

Reuters skup podataka sadrži priče iz novinske agencije Reuters. Korištena je Reuters-21578 verzija korpusa. Korpus je sastavio David Lewis 1987., a dostupan je na http://www.research.att.com/~lewis. Za testiranje metode bitno je da primjeri iz skupa za testiranje ne budu poznati algoritmu i prilikom učenja, jer će rezultati biti nerealistično dobri. Stoga se korpus najčešće dijeli na dva dijela: skup za učenje i skup za testiranje. Da bi rezultati među različitim metodama bili međusobno usporedivi, bitno je da se učenje i testiranje vrši na jednakim skupovima kod svih. Postoji više predefiniranih raspodjela, a ovdje ćemo koristiti ModApte raspodjelu[54].

Ovu raspodjelu sačinjavaju 9603 primjera za učenje i 3299 primjera za testiranje. Kategorija može sadržavati između 1 i 2877 dokumenata u skupu za učenje ili između 1 i 1066 dokumenata u skupu za testiranje. Broj kategorija je 135, međutim, samo se 90 kategorija pojavljuje bar jednom i u skupu za učenje i u skupu za testiranje (Joachims,2001). Od tih 90 mogućih kategorija, mi ćemo koristiti 10 najučestalijih.

Kolekcija dolazi distribuirana u 22 datoteke, zadnja datoteka sadrži 578 dokumenata, dok ostale sadrže po 1000 dokumenata. Datoteke su pisane u SGML[55] formatu.

---

[54] eng. ModApte Split, skraćenica naziva Modified Apte Split

[55] eng. Standard Generalized Markup Language. Jezik za definiranje oznaka unutar dokumenata. Ima specifičan vokabular (oznake za elemente i atribute) i deklariranu sintaksu (gramatiku koja definira hijerarhiju i druge karakteristike). Više informacija može se pronaći na http://xml.coverpages.org/sgml.html.

6.1.2 Baza novinskih članaka Vjesnik

Ovdje je riječ o bazi Vjesnikovih članaka sastavljenoj za potrebe projekta prof. dr.sc. Bojane Dalbelo-Bašić. Bazu je sastavio prof. dr.sc. Marko Tadić sa Odsjeka za lingvistiku Filozofskog fakulteta u Zagrebu, 2004. godine.

Baza sadrži sve novinske članke (Vjesnik između 2000. i 2003. godine, preko 92000 članaka), preuzete iz Hrvatskog nacionalnog korpusa[56]. Dolazi u obliku jedne datoteke pisane u XML[57] formatu. Članci su prikazani u vertikaliziranom obliku[58], stop riječi su uklonjene.

Članci su podijeljeni u 11 kategorija:

- ck - crna kronika
- gl - gledišta (nema u 2003. godini)
- go - gospodarstvo
- ko - komentari (od 2002. godine)
- ku - kultura
- sp - sport
- st - stajališta (samo u 2003. godini)
- td - tema dana
- un - unutarnja politika
- vp - vanjska politika
- zg - zagreb.

Početak svakog članka naznačen je oznakom poput: *<doc type="article" file="vj20000112ck01">,* gdje pojedini dijelovi oznake imaju sljedeće značenje:

- vj - označava pripadnost članka bazi članaka Vjesnik
- 2000 - godina izlaženja članka
- 01 - mjesec izlaženja članka
- 12 - dan izlaženja članka

---

[56] http://www.hnk.ffzg.hr/

[57] eng. Extensible Markup Language, pogledati http://xml.coverpages.org/xml.html. Razvijen je iz SGML-a, vidi bilješku 50. Razlike između SGML-a i XML-a po pitanju funkcionalnosti su gotovo zanemarive (više na http://www.w3.org/TR/NOTE-sgml-xml.html).

[58] Svaka pojavnica jedan redak.

- ck - pripadnost kategoriji (u ovom slučaju radi se o članku iz kategorije crna kronika)
- 01 - redni broj članka te kategorije toga datuma.

6.1.2.1 Baza novinskih članaka Vjesnik HMLv3.0

Riječ je o bazi novinskih članaka Vjesnik iz prethodnog poglavlja na kojoj je provedena morfološka obrada pomoću Hrvatskog morfološkog leksikona verzije 3 - HMLv3.0 - čiji je autor već prije spomenuti prof. dr.sc. Marko Tadić. Hrvatski morfološki leksikon dostupan je na Internetu na adresi [http://www.hnk.ffzg.hr/hml/], a nastao je ručnim generiranjem na osnovi lingvističkog znanja eksperta, što jamči točnost provedene lematizacije.

Proces morfološke obrade pojavnice sastoji se od usporedbi s pojavnicama unutar leksikona. Ukoliko se pojavnica pronađe u leksikonu, dohvaća se njena lema, te se svuda unutar teksta ta pojavnica zamjenjuje svojom lemom. Ovim postupkom broj različitih pojavnica baze članaka Vjesnik se smanjio na oko 75% početnog broja.

6.1.2.2 Baza novinskih članaka Vjesnik AMNv1.0

Riječ je, također, o bazi novinski članaka Vjesnik, morfološka obrada provedena je postupkom Automatske morfološke normalizacije koju je razvio dipl. ing. Jan Šnajder, asistent na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva.

Postupak rada je, u osnovi, sljedeći: za svaku se pojavnicu iz skupa za učenje morfološke normalizacije na temelju pravila određuju svi mogući oblici norme. Zatim se na temelju statističkih svojstava skupa za učenje odredi koji je od tih oblika najvjerojatnije ispravan. Za njega se generira mjesto u leksikonu sa svim oblicima pojavnica koje mogu nastati iz njega. Dakle, Automatska morfološka normalizacija uči normalizirati pojavnice na nekom skupu primjera za učenje[59], stoga radi sa statističkom točnošću.

[59] Taj skup primjera može biti isti onaj koji služi i za učenje klasifikatora.

Postupak morfološke obrade je dalje isti kao i kod ručno generiranog leksikona. Ovom se metodom broj različitih pojavnica u prosjeku smanjio na oko 45% početnog broja pojavnica.

Više informacije o ovoj metodi dostupno je na Internet adresi [http://www.zemris.fer.hr/~jan].

6.1.2.3 Generiranje skupova za učenje i testiranje

Na raspolaganju smo imali veliki broj članaka (92465). Kako taj broj ne odgovara većini algoritama strojnog učenja, a i malo je stvarnih situacija u kojima raspolažemo s tako golemom količinom ulaznih podataka, kao logičan korak činilo se izgraditi skupove za učenje i testiranje s nešto manjim brojem primjera. Reutersova ModApte raspodjela ima 9603 primjera u skupu za učenje. Na temelju toga broja došlo se do 10000 primjera u skupu za učenje i 10000 primjera u skupu za testiranje za bazu članaka Vjesnik.

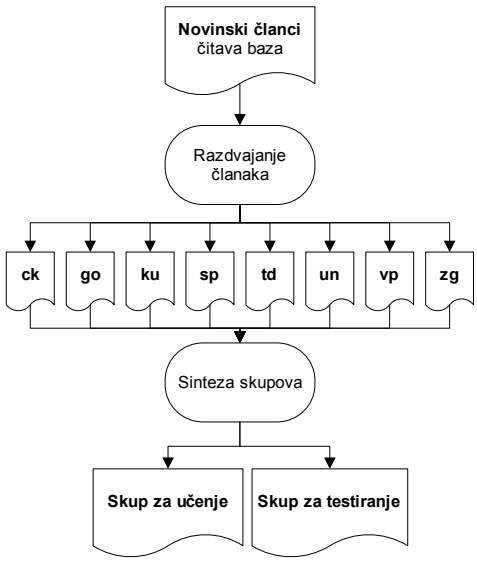
Iz prethodnog razmatranja proizašla je potreba za filtriranjem članaka koji ulaze u tražene skupove. Kako bi se postigla što konzistentnija raspodjela, prvo se rješavalo pitanje kategorija. Kategorije: gledišta, komentari i stajališta isključene su iz sinteze skupova, jer se ne pojavljuju konzistentno unutar kronološkog opsega baze članaka. Zastupljenost preostalih kategorija dana je u tablici 6.1.

naziv kategorije	broj članaka	udio [%]
crna kronika	6954	7.52
gospodarstvo	7837	8.48
kultura	9575	10.36
sport	13952	15.09
tema dana	5375	5.81
unutarnja politika	20352	22.01
vanjska politika	13066	14.13
zagreb	8126	8.79
ostalo	7228	7.82
ukupno	92465	100.00

Tablica 6.1 Zastupljenost kategorija unutar baze Vjesnikovih članaka

Kako bi dobili što kvalitetniji skup za učenje i sukladno što objektivniji skup za testiranje, nužno je bilo obratiti pozornost na vremensku domenu zbog problema

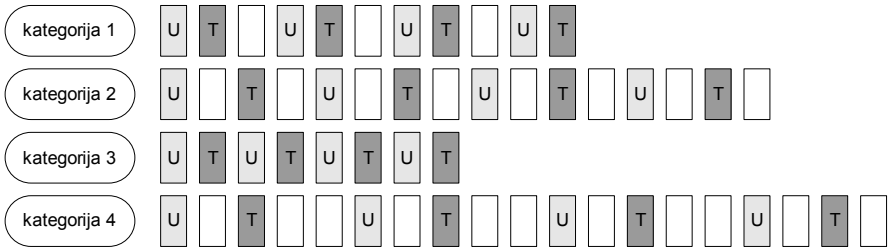
promjene ključnih riječi koje ponajbolje[60] opisuju neku kategoriju. Intuitivno je jasno kako je drugi važan faktor udio pojedine kategorije u skupovima za učenje i testiranje. Kako se, u samom početku, ne bi pravile razlike među kategorijama svaka kategorija sudjeluje s jednakim brojem članaka (njih 10000/8=1250) i u skupu za učenje i u skupu za testiranje. Kako bi se obuhvatila čitava vremenska domena korištena je metoda ravnomjernog uzorkovanja čiji je shematski dijagram prikazan na slici 6.2.



Slika 6.2 Prikaz postupka generiranja skupova za učenje i testiranje

Slika 6.2 prikazuje postupak nastajanja skupova za učenje i testiranje. Članci iz početne baze su raspodijeljeni prema kategoriji kojoj pripadaju u zasebne datoteke zadržavajući kronološki poredak unutar njih.

[60] Ključne riječi koje najbolje opisuju neku kategoriju u određenom trenutku određene su događajima i akterima koji su tada aktualni te stilom pisanja autora članka pa se mogu uvelike razlikovati od ključnih riječi koje bi bile odabrane ukoliko se sagleda čitav vremenski opseg.

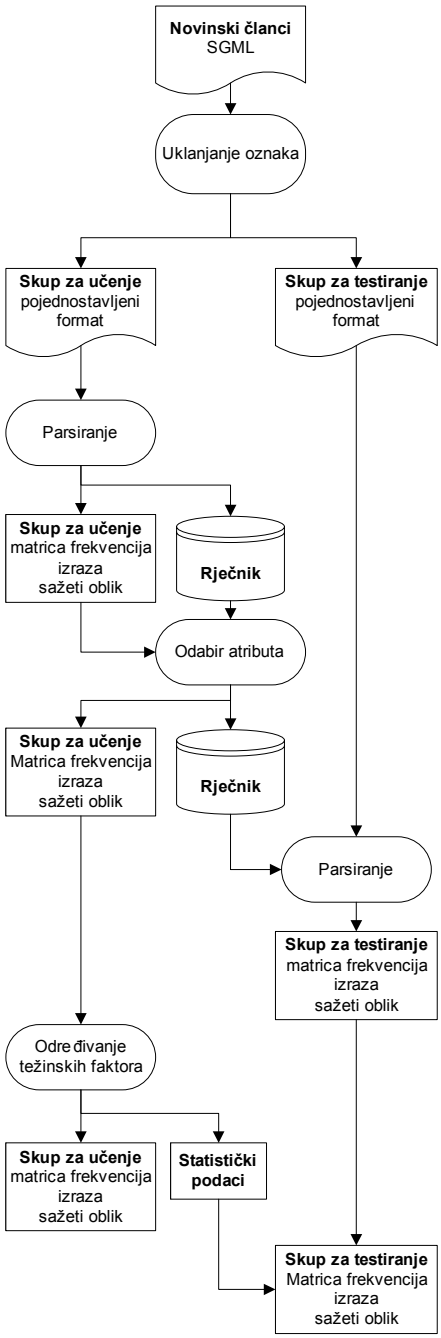


Slika 6.3 Primjer uzorkovanja članaka

Na slici 6.3 primjerom je objašnjen postupak uzorkovanja članaka pri stvaranju skupova. U ovisnosti o broju članaka unutar kategorije te potrebnom broju članaka računaju se koraci pri kojima se odabiru članci (za svaku kategoriju posebno). Na primjeru sa slike 6.3 potrebno je bilo odabrati po 8 članaka iz svake kategorije, od kojih će pola ući u skup za učenje, a druga polovica u skup za testiranje. Slovom U i svjetlijom bojom su označeni oni članci koji su u ušli u skup za učenje, a slovom T i tamnijom bojom članci iz skupa za testiranje.

U procesu stvaranja skupova, prije svakog dodavanja članka skupu, metodom slučajnog odabira određuje se kategorija iz koje će taj članak doći. Ovaj postupak traje sve dok se iz svake kategorije ne odabere traženi broj članaka. Razlog korištenja ovakvog postupka je taj što performanse mnogih algoritama ovise o redoslijedu predočavanja primjera tijekom postupka učenja.

6.2 Predprocesiranje



Slika 6.4 Detaljna shema predprocesiranja

Ulaz u fazu predprocesiranja sačinjava korpus novinskih članaka u određenom formatu. Izlaz je dan u obliku dviju matrica: jedna predstavlja skup za učenje, a druga predstavlja skup za testiranje. Cijela faza predprocesiranja se obavlja u nekoliko koraka. Prvo je potrebno iz formata u kojem su pohranjeni članci izdvojiti one informacije koje nas zanimaju, zatim kreirati skupovi primjera za učenje i testiranje koji su zapisani u pojednostavljenom formatu. Datoteka sa skupom za učenje parsira se kako bi se odvojile pojavnice na temelju kojih se gradi matrica frekvencija izraza[61] te rječnik. Na temelju matrice frekvencija izraza nekom od metoda odabire se podskup atributa što rezultira smanjenjem dimenzije matrice frekvencija izraza i smanjenjem opsega rječnika. Sada se vrši parsiranje i primjera iz skupa za testiranje tako da se evidentira samo pojavljivanje riječi iz rječnika. Rezultat je matrica frekvencija izraza skupa za testiranje. Sljedeći korak je dodjeljivanje težinskih faktora matricama frekvencija izraza koje se vrši na temelju rezultata statističke obrade nad matricom primjera za učenje. Ovo je krajnji rezultat predprocesiranja i ulaz u postupak učenja.

6.2.1 Struktura dokumenata i pojednostavljeni oblik

Iako su nazivom različiti XML i SGML, formati na ovoj razini funkcionalnosti veoma su slični. Oba formata definiraju oznake koje odvajaju članke unutar datoteke i oznake koje odvajaju neke posebne informacije o članku unutar samog članka, poput naslova, imena autora i slično. U postupku kategorizacije nijedna nam od tih informacija nije potrebna, a ni prikladna za proces odvajanja pojavnica. Stoga je ove formate potrebno prevesti u jednostavniji i prikladniji oblik.

6.2.1.1 Svođenje baze Reuters na pojednostavljeni oblik

Na slici 6.5 prikazan je primjer članka iz baze Reuters-21578. Potrebno je članak iz ovoga oblika prevesti u jednostavniji tako da bude pogodniji za parsiranje.

[61] eng. term frequency matrix (izraz - dokument matrica) Redci matrice predstavljaju pojedine izraze, a stupci dokumente. Element matrice označava broj pojavljivanja nekog izraza u nekom dokumentu.

Granice članka određene su oznakom[62] REUTERS. Ova oznaka sadrži nekoliko atributa, a nama su bitni samo oni koji određuju pripadnost članka ModApte raspodjeli, odnosno skupu za učenje ili skupu za testiranje unutar te raspodjele[63].

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="5544" NEWID="1">
<DATE>26-FEB-1987 15:01:01.79</DATE>
<TOPICS><D>cocoa</D></TOPICS>
<PLACES><D>el-salvador</D><D>usa</D><D>uruguay</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&#5;&#5;&#5;C T
&#22;&#22;&#1;f0704&#31;reute
u f BC-BAHIA-COCOA-REVIEW   02-26 0105</UNKNOWN>
<TEXT>&#2;
<TITLE>BAHIA COCOA REVIEW</TITLE>
<DATELINE>    SALVADOR, Feb 26 - </DATELINE><BODY>Showers
continued throughout the week in
the Bahia cocoa zone, alleviating the drought since early
January and improving prospects for the coming temporao,
although normal humidity levels have not been restored,
Comissaria Smith said in its weekly review.
...
    Final figures for the period to February 28 are expected to
be published by the Brazilian Cocoa Trade Commission after
carnival which ends midday on February 27.
 Reuter
&#3;</BODY></TEXT>
</REUTERS>
```

Slika 6.5 Primjer članka iz baze Reuters

```
<REUTERS -1>
BAHIA COCOA REVIEW
BAHIA COCOA REVIEW
Showers continued throughout the week in
the Bahia cocoa zone alleviating the drought since early
january and improving prospects for the coming temporao
although normal humidity levels have not been restored
Comissaria Smith said in its weekly review
...
    Final figures for the period to February BROJ are expected to
be published by the Brazilian Cocoa Trade Commission after
carnival which ends midday on February BROJ
 Reuter
```

Slika 6.6 Pojednostavljeni oblik članaka iz baze Reuters

[62] eng. tag. Većina oznaka dolazi u parovima: <oznaka> predstavlja početak, a </oznaka> prestanak djelovanja pojedine oznake.

[63] Više informacija o pripadnosti dokumenta pojedinoj raspodjeli može se naći u datoteci koja dolazi uz bazu članaka Reuters-21578, a opisuje njenu strukturu.

Ukoliko se među oznakama TOPIC nalazi naziv željene kategorije, onda joj članak i pripada[64]. Kako bi istaknuli veću važnost teksta u naslovu članka, tekst između oznaka TITLE ponovimo dva ili više puta i pridružimo ostatku članka označenom s BODY. Oznake TITLE i BODY uklonimo, a njihov sadržaj zadržimo. Ostale oznake, osim REUTERS, uklanjamo zajedno sa sadržajem.

Odstranjujemo sve interpunkcijske znakove iz teksta jer smatramo da ne nose informaciju o kategoriji teksta. Brojeve zamjenjujemo jednim izrazom koji predstavlja klasu, jer pretpostavljamo kako više informacije o kategoriji teksta nosi podatak o učestalosti brojeva u tekstu nego podaci o točnim vrijednostima koje se pojavljuju. Tekst članka prikazujemo kao niz riječi odvojenih praznim znakovima.

6.2.1.2 Svođenje baze Vjesnik na pojednostavljeni oblik

```
<doc type="article" file="vj20000112ck01">
<head type="na">
slučaju
zagrebačke
mafije
saslušat
300
svjedoka
</head>
<p>
<b>
zagreb
11
siječnja
</b>
nastavku
istrage
slučaju
nikice
...
saslušati
tajno
imena
zna
</p>
</doc>
```

Slika 6.7 Primjer članka iz baze Vjesnik

Slika 6.7 prikazuje primjer članka iz baze vjesnik. Očito je kako se radi o vertikaliziranom načinu zapisivanja. Granice članka određene su parom oznaka

[64] Pripadnost kategoriji označavamo pomoću vrijednosti +1 na mjestu atributa oznake REUTERS, ne pripadanje članka traženoj kategoriji označavamo s -1 na istom tom mjestu (vidi slika 6.6).

<doc> i </doc> iz kojih se izvlači informacija o pripadnosti određenoj kategoriji[65]. Među ostalim oznakama koje se pojavljuju u ovom formatu zapisivanja valja istaknuti <head> i </head> koje ograđuju naslove i podnaslove unutar članka. One su bile od iznimne važnosti prilikom istraživanja količine informacije koju naslovi i podnaslovi nose o članku[66]. Ostale oznake samo se uklanjaju iz članka ne dirajući tekst koji ograđuju. Rezultat takve obrade dan je na slici 6.8, gdje je vidljivo kako početak članka označava oznaka <vjesnik>, zajedno s atributom koji označava pripadnost ili ne pripadnost promatranoj kategoriji, nakon čega slijede riječi koje se pojavljuju u tekstu.

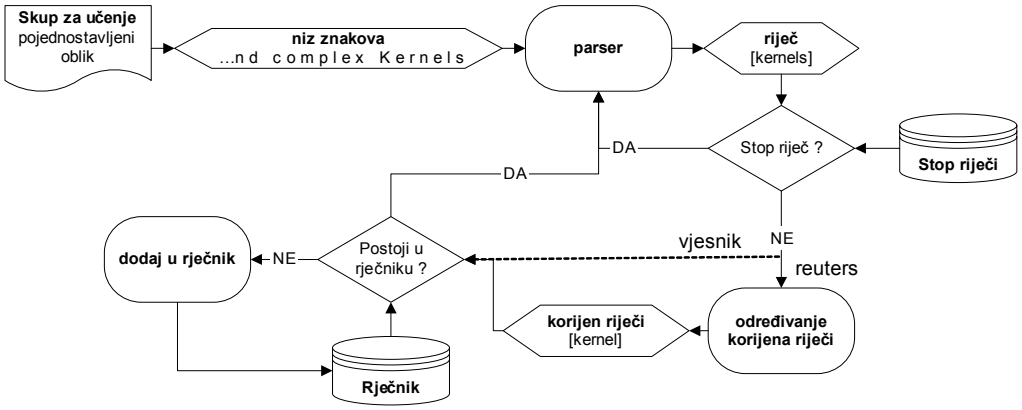
```
<vjesnik -1>
slučaju
zagrebačke
mafije
saslušat
svjedoka
zagreb
siječnja
nastavku
istrage
slučaju
nikice
...
saslušati
tajno
imena
zna
```

Slika 6.8 Pojednostavljeni oblik članka sa slike 6.7

[65] Značenje atributa iz oznake <doc>, a time i postupak određivanja kategorije članka, opisano je u poglavlju 6.1.2.

[66] Promatrana je ovisnost efikasnosti klasifikatora o broju ponavljanja naslova i podnaslova unutar članka.

6.2.2 Generiranje rječnika



Slika 6.9 Shema stvaranja rječnika

Na slici 6.9 dana je shema procesa stvaranja rječnika od podataka iz skupa za učenje.

Bitno je naglasiti kako su obje baze, Vjesnik i Reuters, pojednostavljivanjem svedene na identičan oblik s obzirom na rad parsera. Prilikom parsiranja, sva se slova, ukoliko već nisu, transformiraju u odgovarajuća mala slova[67]. Kada parser detektira riječ, provjerava da li je sadržana u listi stop riječi. Ukoliko je, zanemaruje se, te se od parsera traži da dohvati novu riječ, a ukoliko nije, prolazi daljnju obradu. Ako radimo s bazom Reuters, Porterovim algoritmom će se odrediti korijen riječi, a ukoliko se radi o bazi Vjesnik, riječ će se samo proslijediti dalje[68].

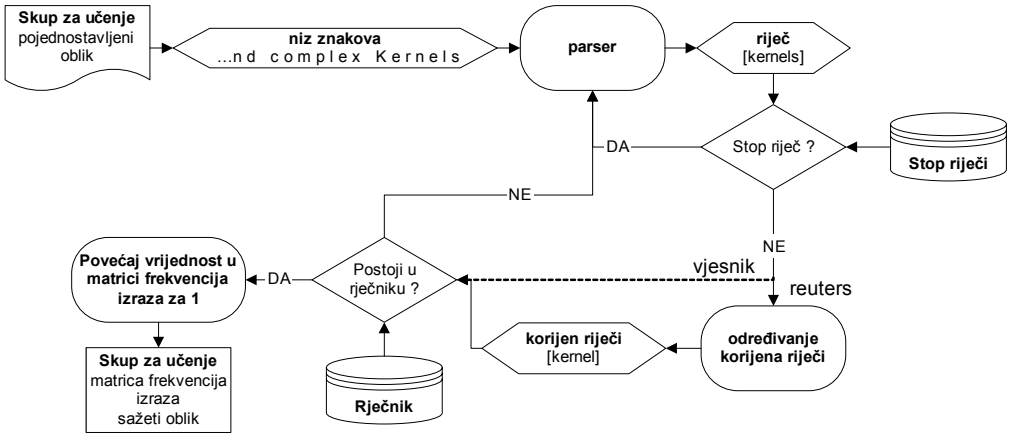
Dobivena riječ se uspoređuje s postojećima u rječniku. Ako se već ne nalazi u rječniku, dodaje se; inače se zanemaruje te se prelazi na obradu sljedeće riječi.

---

[67] Razlog ovome je pretpostavka kako informacija koju nosi činjenica da je neka riječ pisana velikim ili malim slovima nije dovoljno zanimljiva da bi opravdala veliko povećanje dimenzionalnosti zbog puno većeg broja pojavnica koji bi se detektirao.

[68] Ovo je posljedica nedostupnosti funkcije koja bi određivala korijene ili lemu hrvatskim riječima. Ovaj nedostatak izbjegnut je radom s dvije inačice baze Vjesnik: Vjesnik HMLv3.0 i Vjesnik AMNv1.0 koje su prošle postupak morfološke obrade. Vidi poglavlja 6.1.2.1 i 6.1.2.2.

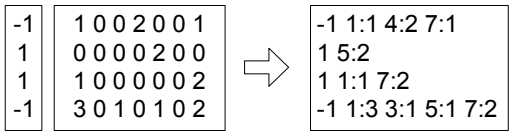
6.2.3 Generiranje matrice za učenje



Slika 6.10 Shema procesa generiranja matrice za učenje

Slika 6.10 prikazuje shemu procesa generiranja matrice frekvencija izraza za učenje. Proces je vrlo sličan onome pri kojem generiramo rječnik. Razlika među bazama Vjesnik i Reuters i dalje je u nepostojanju postupka određivanja korijena riječi za bazu Vjesnik. Ukoliko riječ postoji u rječniku, dobavlja se njezin indeks koji zapravo označava redak matrice frekvencija koji pripada danoj riječi. Za jedan se uvećava element polja matrice kojemu je redak jednak indeksu riječi iz rječnika, a stupac indeksu dokumenta koji se trenutno obrađuje.

Matrica se u datoteku zapisuje u sažetom libsvm[69] formatu.



Slika 6.11 Primjer preslikavanja u libsvm format

Na slici 6.11 dan je primjer zapisivanja matrice u takvom formatu. Očito je kako se u jednoj datoteci pohranjuju i oznaka kategorije i polja matrice. Ovaj je format vrlo ekonomičan, jer iz Zipfovog zakona[70] slijedi kako su vektori koji sačinjavaju matricu,

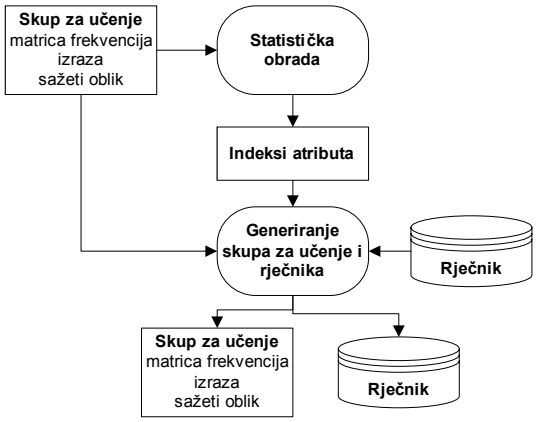
---

[69] libsvm - eng. A Library for Support Vector Machines. Biblioteka funkcija za modeliranje klasifikatora metodom potpornih vektora. Dostupna na http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[70] Vidi poglavlje 3.1

a prikazuju dokumente frekvencijama pojavljivanja izraza iz rječnika, vrlo slabo popunjeni.

6.2.4 Odabir atributa



Slika 6.12 Shematski prikaz procesa odabira atributa

Metode procesa odabira atributa već su objašnjene[71]. Ovdje su implementirane dvije metode: metoda odabira atributa na temelju frekvencije dokumenata - DF i metoda na temelju informacijske dobiti - IG. Postupak je isti kod obje metode i bit će objašnjen na temelju shematskog prikaza sa slike 6.12.

Na temelju matrice skupa za učenje, vrednuju se atributi prema nekom od kriterija (IG ili DF). Atributi koji vrijednošću prijeđu određeni prag, tvore listu indeksa atributa. Na temelju ove liste i rječnika stvara se novi rječnik izborom samo onih riječi čiji se indeksi pojavljuju u toj listi. Iz matrice skupa za učenje stvara se nova matrica skupa za učenje prepisivanjem samo onih redaka čiji se indeksi pojavljuju u listi indeksa atributa.

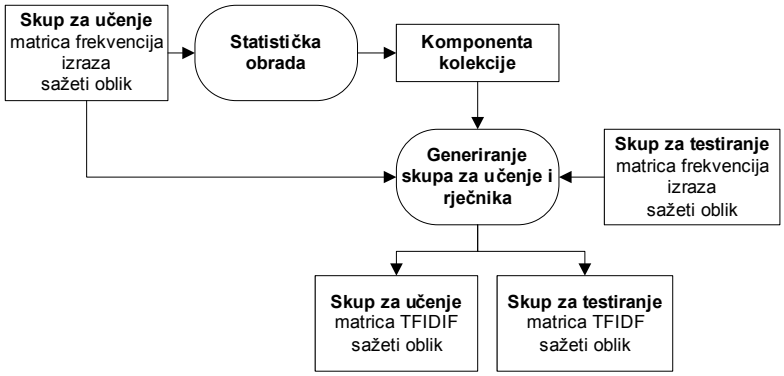
6.2.5 Generiranje matrice za testiranje

Matrica za testiranje kreira se prema shemi sa slike 6.10, uz opasku da naziv *skup za učenje* treba svuda zamijeniti nazivom *skup za testiranje*.

[71] Pogledati poglavlje pod naslovom Odabir atributa.

Matrica za testiranje gradi se na temelju rječnika koji je nastao nakon procesa odabira atributa. Ovim su načinom obje matrice prikazane posredstvom istih atributa.

6.2.6 Dodjeljivanje težinskih faktora elementima matrica



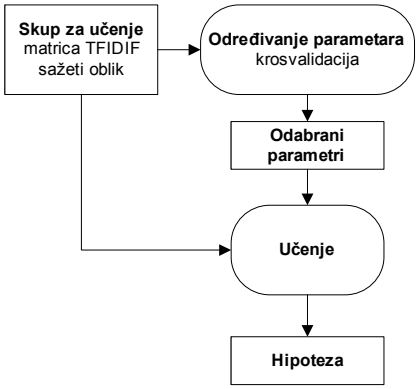
Slika 6.13 Shematski prikaz procesa dodjeljivanja težinskih faktora

Težinski faktori koji se dodjeljuju elementima matrica sastoje se od tri dijela[72]: komponente dokumenta, komponente kolekcije i normalizacijske komponente. Komponentu kolekcije skupa za učenje koristi i skup za testiranje, stoga ju je potrebno izračunati i pohraniti. Druge dvije komponente generiraju se posebno za svaki skup. Dakle, na temelju ovih triju komponenti generiraju se matrice s težinskim faktorima.

Svaka od komponenti ima više varijacija i ovisno o njima formira se prikaz skupa. Ovdje je implementirana najčešća kombinacija koja rezultira TFIDF prikazom matrica.

[72] Vidi poglavlje 3.4

6.3 Učenje



Slika 6.14 Shematski prikaz procesa učenja

Slika 6.14 daje shematski prikaz procesa učenja. Očito je kako je ulaz u ovaj proces definiran matricom skupa za učenje, a izlaz je naučena hipoteza koja vrši klasifikaciju dokumenata.

Bitan proces u postupku učenja je i odabir kvalitetnih parametara učenja, jer su oni ti koji će, uz odabir odgovarajućeg tipa klasifikatora, dati kvalitetne rezultate.

6.3.1 Biblioteka funkcija LIBSVM

Prilikom implementacije metode potpornih vektora, koristio sam se javno dostupnom bibliotekom funkcija LIBSVM[73], pisanih u programskom jeziku C. Detalji o njoj mogu se pronaći na [http://www.csie.ntu.edu.tw/~cjlin/libsvm/].

LIBSVM biblioteka implementira više tipova klasifikatora. Ova implementacija koristi tip nazvan C-SVC koji odgovara klasifikatoru sa slabom marginom u L_1 normi[74].

Podržana je samo RBF jezgrena funkcija, ali kako ona i daje najbolje rezultate[75], to se ne mora smatrati velikim nedostatkom.

---

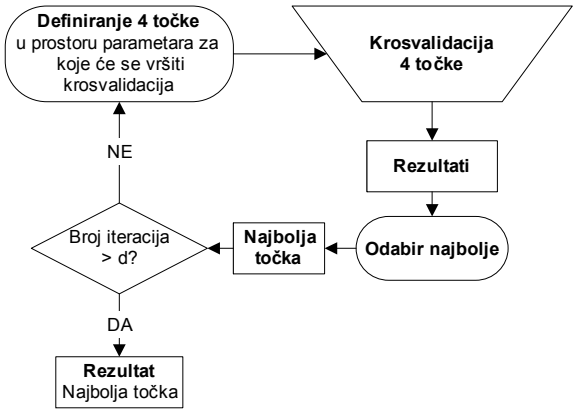
[73] Autori ove biblioteke su Chih-Chung Chang i Chih-Jen Lin, Department of Computer Science and Information Engineering, National Taiwan University.

[74] Vidi poglavlje 5.2.2

[75] Ovo vrijedi izuzmemo li specijalizirane jezgrene funkcije.

Oko ove biblioteke izgrađeno je sučelje koje implementira postupak odabira parametara krosvalidacijom te postupak vrednovanje rezultata.

6.3.2 Određivanje parametara učenja

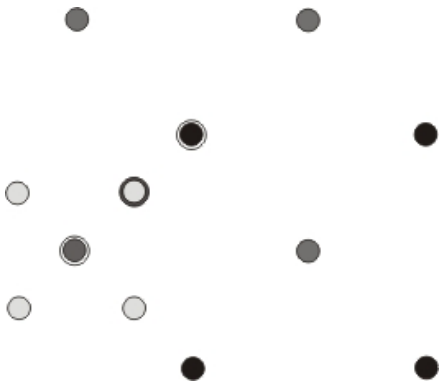


Slika 6.15 Grubi shematski prikaz odabira parametara učenja

Kako prostor razapet parametrima sadrži beskonačno mnogo točaka, a i svako temeljito pretraživanje zahtjeva mnogo vremena, implementirao sam heurističku metodu ugrubo prikazanu na slici 6.15.

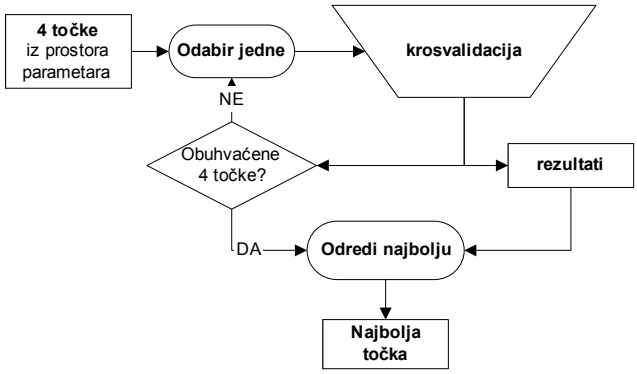
Korisnik definira četiri početne točke u prostoru parametara, među njima se odabere jedna za koju se krosvalidacijom utvrdi kako daje najbolja[76] rješenja. U okolini te točke odrede se nove četiri za koje se onda opet krosvalidacijom dobiju rezultati. Bira se opet najbolja točka uzevši u obzir i dotad najbolju, te se postupak nastavlja određivanjem četiri nove točke dok se ne napravi određeni broj iteracija d.

[76] Pod najboljim se smatra ono rješenje koje postigne najviši rezultat u smislu F_1 mjere.



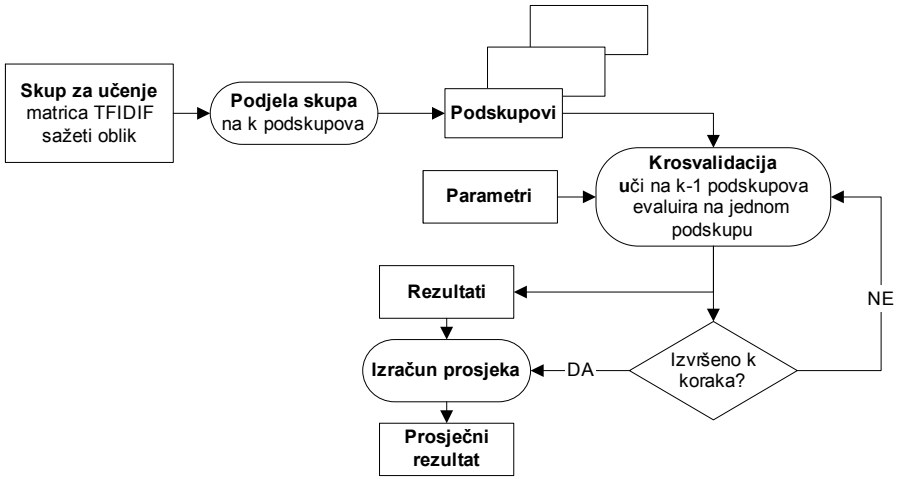
Slika 6.16 Primjer odabira parametara

Na slici 6.16 dan je primjer odabira parametara za d=3. Najtamnijim bojom označene su početne četiri točke, malo svjetlijom sljedeće četiri, dok su najsvjetlijom bojom označene posljednje četiri točke. U svakoj iteraciji dodatno je zaokružena točka koja daje najbolje rezultate u postupku krosvalidacije.



Slika 6.17 Shematski prikaz postupka krosvalidacija 4 točke

Na slici 6.17 dana je shema postupka "krosvalidacije 4 točke". Smisao je jednostavan, od četiri točke odabire se ona koja ima najbolje rezultate u postupku krosvalidacije.



Slika 6.18 Shematski prikaz procesa krosvalidacije

Slika 6.18 daje nam shematski prikaz procesa "krosvalidacije". Dakle, matrica skupa za učenje dijeli se po redcima na *k* jednakih dijelova. Smisao je da se uči na *k-1* dijelova, a rezultati vrednuju na preostalom jednom dijelu. Ovaj se postupak ponavlja *k* puta. Kao rezultat uzima se prosječna vrijednost rezultata postignutih u izvršenim koracima.

6.3.3 Učenje

Učenje se obavlja jednostavnim pozivom funkcije iz libsvm biblioteke. Kao parametri učenja, funkciji se predaju rezultati iz postupka odabira parametara. Učenje se obavlja na cijelom skupu za učenje, a rezultat je datoteka koja sadrži podatke što grade naučenu hipotezu.

6.4 Testiranje i vrednovanje rezultata

U ovom poglavlju prikazani su i interpretirani rezultati eksperimenata izvršenih nad bazama Reuters i Vjesnik. Mjerenja su podijeljena u više kategorija. Prva kategorija mjerenja daje statističke podatke o dostupnim bazama podataka. Druga kategorija mjerenja ima zadaću da prikaže efikasnost implementiranog klasifikatora, dok je treća napravljena s ciljem vrednovanja rada nekih jezičnih metoda kojima se nastojala smanjiti dimenzionalnost prostora u bazi članaka Vjesnik.

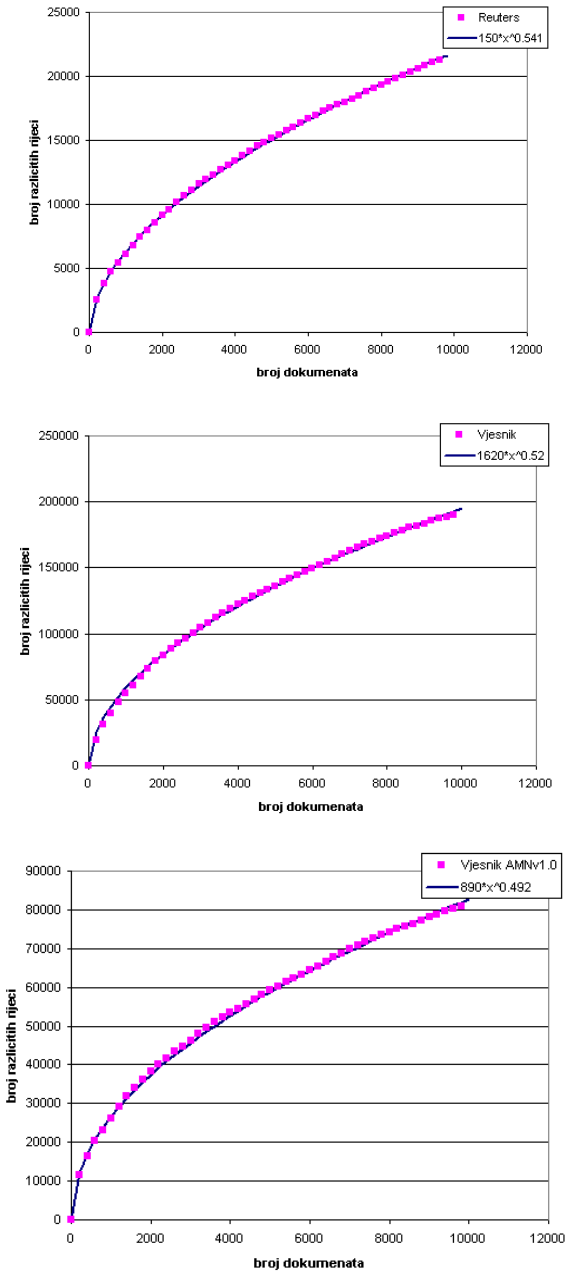
6.4.1 Statistička svojstva teksta

Kako bi bolje razumjeli prirodu problema u susretu s ovim bazama podataka, činilo se neophodnim provesti i neka statistička istraživanja. Ranije su već navedena neka svojstva teksta koja bi trebala vrijediti neovisno o jeziku i vrsti teksta. Riječ je o broju različitih riječi koje se pojavljuju u tekstu u ovisnosti o broju dokumenata tj. Heapsovom zakonu, te frekventnosti pojavljivanja riječi iz teksta ili Zipfovom zakonu.

6.4.1.1 Heapsov zakon[77]

Ovisnost broja atributa o broju dokumenata, indikativan je pokazatelj rasta dimenzionalnosti prostora učenja s povećanjem broja obuhvaćenih dokumenata. Ovaj pokazatelj ima važnu ulogu pri konstrukciji skupova za učenje i testiranje, jer služi kao protuteža nastojanju da se što više poveća broj dokumenata. Ovime se održava ravnoteža između uspješnost klasifikatora i brzine rada.

[77] Pogledati pod 7.



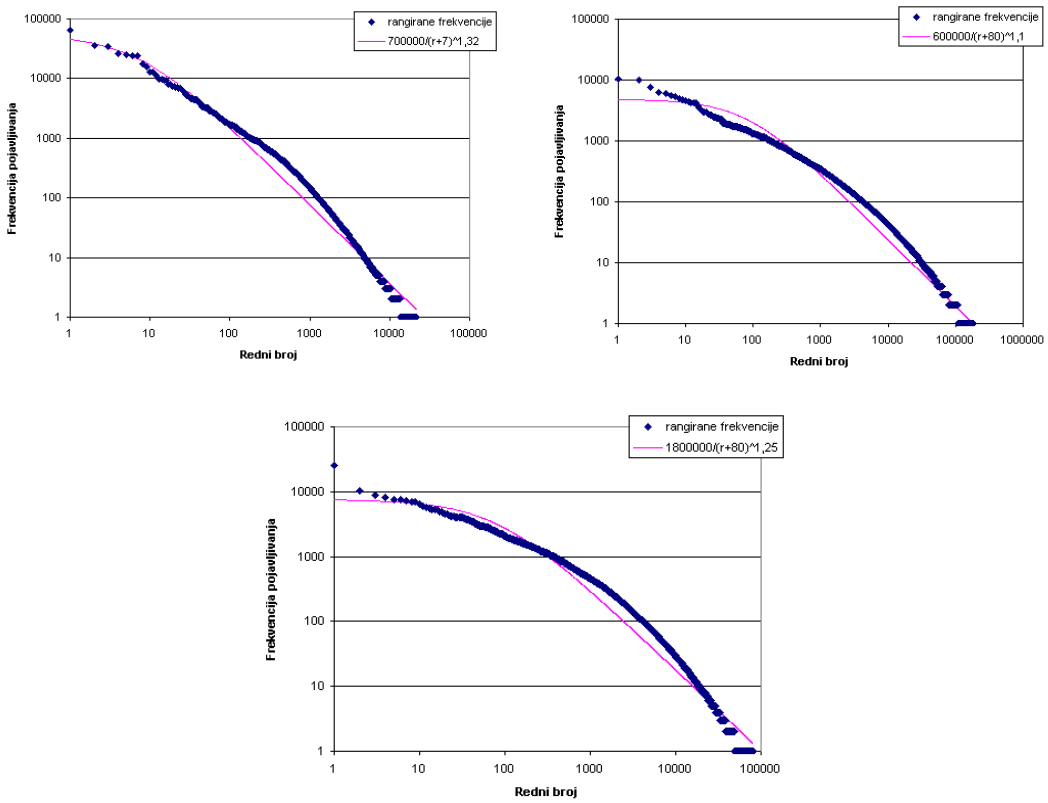
Slika 6.19 Ovisnost broja različitih riječi o broju dokumenata u bazama članaka Reuters, Vjesnik i Vjesnik AMNv1.0

Na slici 6.19 prikazane su ovisnosti broja različitih riječi o broju dokumenata. Jasno se vidi kako slijede Heapsov zakon dan izrazom (6.1).

$$N = k \cdot s^{\beta} \qquad (6.1)$$

Gdje je N broj različitih riječi, s oznaka za broj dokumenata, a k i β konstante. Poznavajući ovu ovisnost, lako je s vrlo velikom točnošću predvidjeti broj atributa s kojim će klasifikator morati raditi.

6.4.1.2 Zipfov zakon



Slika 6.20 Primjena Zipfovog zakona na baze Reuters, Vjesnik i Vjesnik AMNv1.0

Primjeri sa slike 6.20 prikazuju slaganja ulaznih podataka sa Zipfovim zakonom čiji je iskaz dan u poglavlju 3.1. Također je navedeno kako se eksperimentalni rezultati bolje slažu s općenitijim oblikom Zipfove raspodjele (izraz 3.1), tj. s Mandelbrotovom raspodjelom (izraz 3.2). Na slici 6.20 određeni su parametri Mandelbrotove raspodjele tako da bi što bolje odgovarali podacima iz baza Reuters, Vjesnik i Vjesnik AMNv1.0.

Svrha ovoga eksperimenta je što zornije prikazati razlog zbog kojeg je izraz-dokument matrica tako slabo popunjena. Iz gornjih se primjera jasno vidi kako se samo mali broj riječi pojavljuje jako često, a većinu od tih riječi ionako čine stop

riječi koje ne ulaze u izraz-dokument matricu, dok se gotovo polovica riječi pojavi samo jednom.

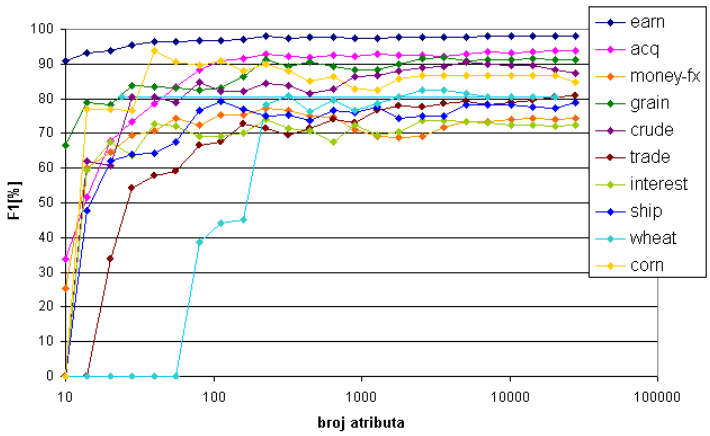
Također se jasno može vidjeti i kolika je redukcija dimenzionalnosti prostora atributa uklonimo li one riječi čija je frekvencija pojavljivanja manja od neke unaprijed definirane vrijednosti.

6.4.2 Uspješnost u ovisnosti o broju atributa

Efikasna redukcija dimenzionalnosti prostora atributa jedno je od osnovnih nastojanja u području strojnog učenja. Implementirane su metode odabira atributa na osnovi količine informacijske dobiti[78] - IG te na osnovi frekvencije dokumenata[79] - DF.

6.4.2.1 Rezultati za bazu članaka Reuters

Slika 6.21 prikazuje ovisnost mjere F1 o broju atributa odabranih na temelju mjere IG, za sve kategorije baze Reuters.

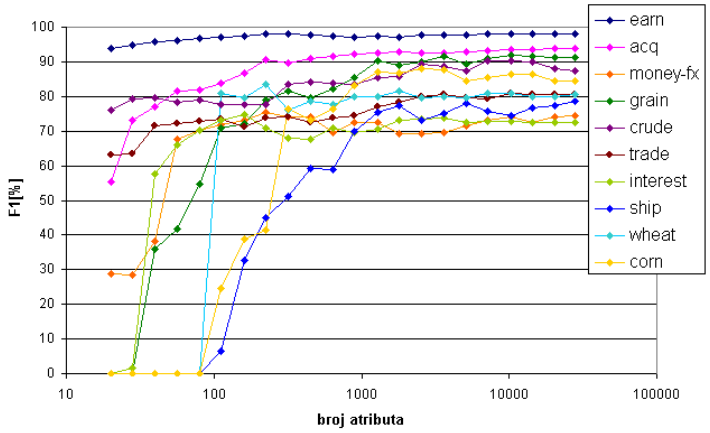


Slika 6.21 Ovisnost mjere F1 o broju atributa po kategorijama baze članaka Reuters. Atributi su birani na osnovi količine informacijske dobiti - IG (γ=0.01, C=1000)

[78] Pogledati poglavlje 3.2.3

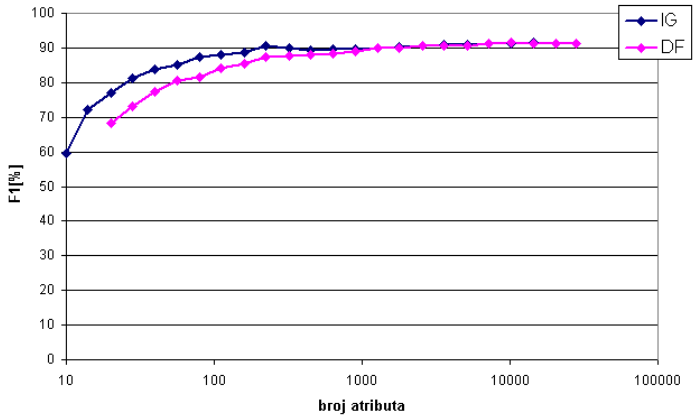
[79] Pogledati poglavlje 3.2.2

Sve kategorije, uz neke manje oscilacije, pokazuju tendenciju povećanja uspješnosti s povećanjem broja atributa. Isto se lako da uočiti kako svoj maksimum postižu i mnogo ranije nego što se dostigne maksimalan broj atributa.



Slika 6.22 Ovisnost mjere F1 o broju atributa po kategorijama baze članaka Reuters. Atributi su birani na osnovi frekvencije dokumenata - DF (γ=0.01, C=1000)

Slika 6.22 pokazuje isto što i slika 6.21, uz razliku što se za kriterij odabira atributa koristila mjera DF. Oblik krivulje je i dalje zadržan, no maksimumi uspješnosti dostižu se nešto kasnije.



Slika 6.23 Ovisnost mikrousrednjene mjere F1 o broju atributa (biranih metodama IG i DF) u bazi članaka Reuters (γ=0.01, C=1000)

Slika 6.23 sumira rezultate s prethodne dvije slike prikazujući mikrousrednjenu[80] vrijednost mjere F_1. Sada se jasno vidi kako je informacijska dobit mnogo bolji kriterij za odabir atributa učenja, jer se zadovoljavajući rezultati postižu već na oko 400 atributa, dok se kod mjere DF oni postižu tek negdje na oko 2000 atributa.

U tablici 6.2 navedeni su rezultati eksperimenata iz (Joachims,2001) zajedno s rezultatima dobivenim u okvirima ovoga diplomskog rada. Iako postoje neke razlike[81], u provedbama eksperimenata okvirno se može zaključiti kako su ovdje dobiveni rezultati u rangu s onima iz (Joachims,2001).

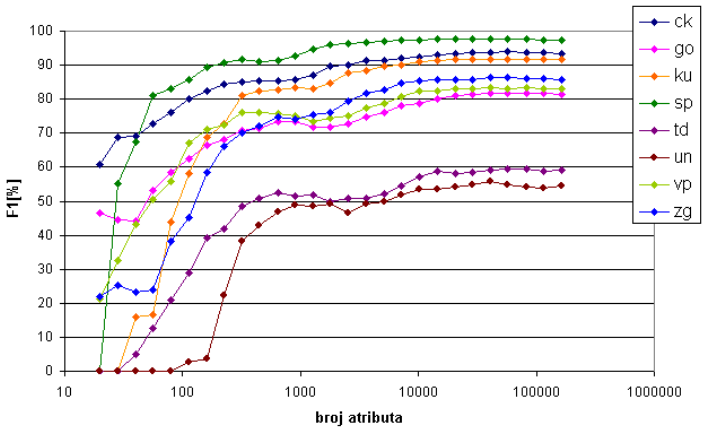
	(Joachims,2001) γ=0,01	γ=0,01 C=1000
earn	98,1	98,1
acq	94,7	93,7
money-fx	74,3	74,5
grain	93,4	91,8
crude	88,7	90,6
trade	76,6	80,7
interest	69,1	73,8
ship	85,8	78,8
wheat	82,4	82,4
corn	84,6	93,7

Tablica 6.2 Usporedba rezultata (točke izjednačenja) navedenih u (Joachims,2001) uz parametre: broj atributa=1000, γ=0,01 s rezultatima dobivenim u okvirima ovoga rada uz parametre: C= 1000, γ=0,01.

[80] Pogledati poglavlje 4.4

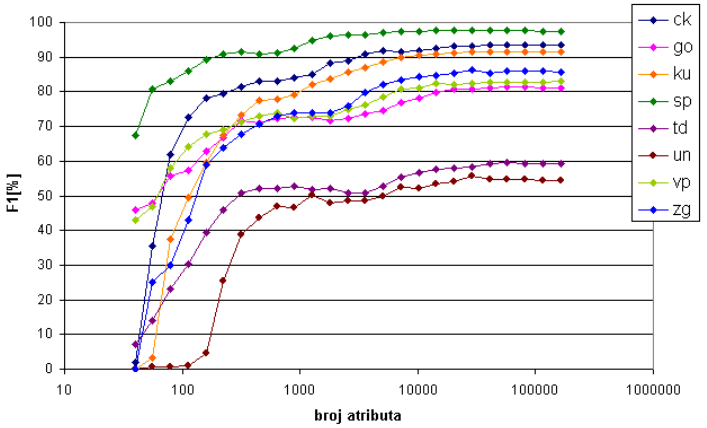
[81] U (Joachims,2001) nepromjenjivim se držao broj atributa, dok se u ovom radu nepromjenjivim držao iznos parametra C.

6.4.2.2 Rezultati za bazu članaka Vjesnik



Slika 6.24 Ovisnost mjere F1 o broju atributa po kategorijama baze članaka Vjesnik. Atributi su birani na osnovi količine informacijske dobiti (IG, γ=0.01, C=1000)

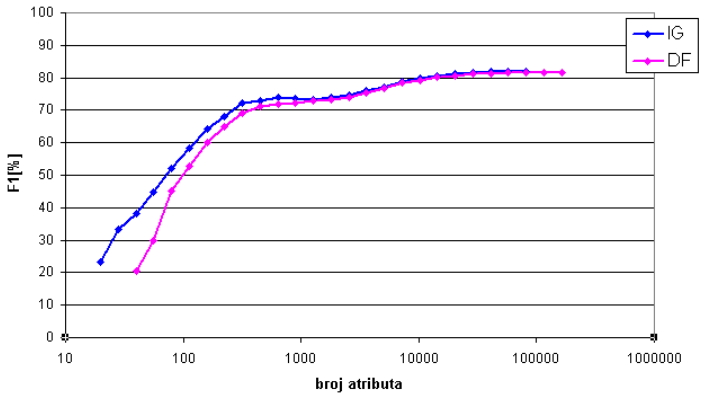
Slika 6.24 prikazuje ovisnost mjere F_1 o broju atributa za kategorije baze članaka Vjesnik. Rezultati su slični rezultatima za bazu Reuters (slika 6.21), no čini se kako se kod Vjesnikove baze informativni atributi odabiru u dvije faze[82].



Slika 6.25 Ovisnost mjere F1 o broju atributa po kategorijama baze članaka Vjesnik. Atributi su birani na osnovi frekvencije dokumenata (DF, γ=0.01, C=1000)

[82] Ove faze su, na slici, označene dijelovima krivulje s velikim gradijentom.

Slika 6.25 prikazuje porast uspješnosti selekcijom sve većeg broja atributa na temelju frekvencije dokumenata. Krivulje su vrlo slične onima sa slike 6.25, tek su malo pomaknute udesno.



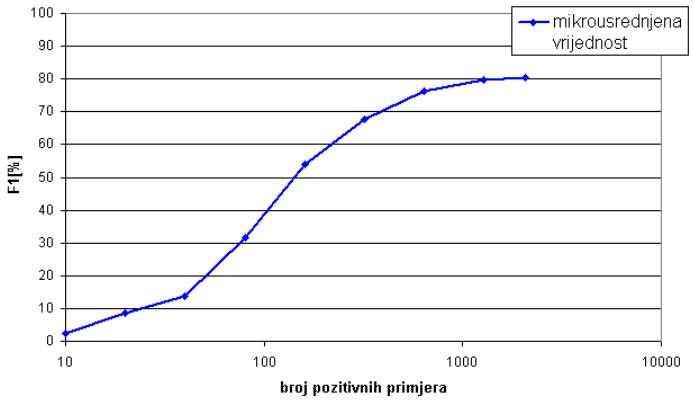
Slika 6.26 Ovisnost mikrousrednjene mjere F1 o broju atributa (biranih metodama IG i DF) u bazi članaka Reuters (γ=0.01, C=1000)

Informacijska se dobit i na bazi članaka Vjesnik pokazuje kao bolji kriterij odabira atributa od mjere DF. Razlike su najveće na samom početku, dok se kasnije, za oko 10000 atributa, krivulje spajaju u jednu, jer su svi informativni atributi već obuhvaćeni. Slika 6.26 mnogo zornije prikazuje dva područja intenzivnog rasta uspješnosti klasifikatora. Čini se kako obje metode na području između 300-tog i 2000-tog atributa odabiru neinformativne atribute. Daljnja bi istraživanja trebala ispitati neke druge kriterije odabira atributa kako bi se razjasnilo, da li je uzrok odabir kriterija ili nešto drugo. Rješenje ovog problema obećava vrlo dobru klasifikaciju već pri 1000 atributa, što je za red veličina bolje od 10000 atributa gdje se sada postižu zadovoljavajući rezultati.

6.4.3 Uspješnost u ovisnosti o broju primjera za učenje

Ova mjerenja imaju za cilj pokazati kako veličina i struktura skupa članaka za učenje utječu na rezultate klasifikatora. Mjerenja su podijeljena na dva dijela. Prvi dio prikazuje ovisnost uspješnosti o broju pozitivnih primjera unutar skupa za učenje, dok drugi dio prikazuje ovisnost uspješnosti o veličini cjelokupnog skupa za učenje.

6.4.3.1 Ovisnost o broju pozitivnih primjera



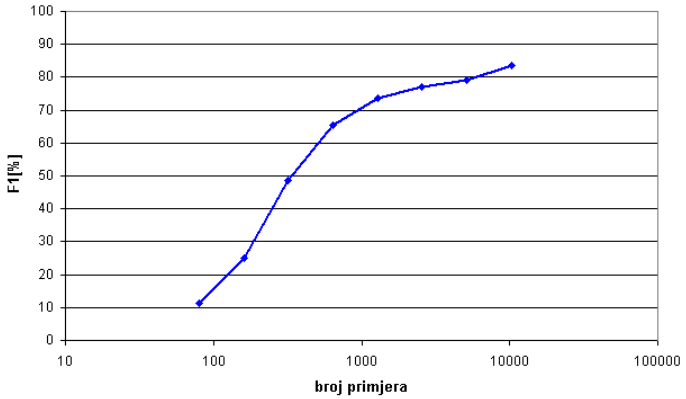
Slika 6.27 Ovisnost mikrousrednjene mjere F1 o broju pozitivnih primjera za učenje u bazi članaka Vjesnik (broj članaka promatrane kategorije varira, dok broj članaka po ostalim kategorijama iznosi 1250, 10000 atributa,IG, γ=0.01, C=1000)

Broj članaka po svim kategorijama koje tvore skup za učenje drži se konstantnim i iznosi 1250, osim promatrane kategorije za koju se varira. Broj atributa također se drži konstantnim na 10000, a odabire ih se metodom IG. Ovo mjerenje bilo je moguće izvršiti jer smo na raspolaganju imali jako velik broj članaka[83], te smo si mogli priuštiti luksuz kreiranja vlastitih skupova na način koji je nama najviše odgovarao. Rezultati pokazuju ono što se moglo i naslutiti: uspješnost

[83] Pogledati poglavlje 6.1.2.

u početku vrlo brzo raste[84], no taj gradijent se s povećanjem broja primjera sve više smanjuje. Ovi rezultati ukazuju na potrebu velikog broja pozitivnih primjera koji će dobro opisati kategoriju, također je očito kako nema smisla broj primjera povećavati unedogled, jer svaki novi primjer donosi sve manju količinu informacije.

6.4.3.2 Ovisnost o ukupnom broju primjera



Slika 6.28 Ovisnost mikrousrednjene mjere F1 o broju primjera u skupu za učenje, baza članaka Vjesnik (atributa 10000, IG, γ=0.01, C=1000)

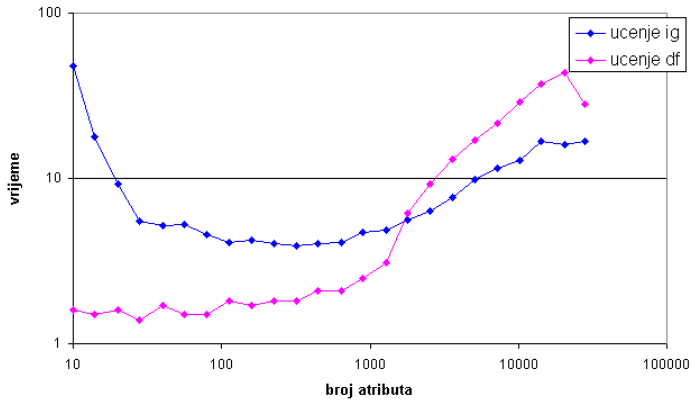
Ovaj se eksperiment odvijao na način da su zadržani relativni odnosi po broju primjera među kategorijama, a mijenjao se ukupan broj primjera. Izgledom je ova krivulja vrlo slična onoj sa slike 6.27. Povećanje uspješnosti, u početku, vrlo brzo raste s porastom broja primjera, no vremenom je gradijent sve manji. Jasno je kako se radi o ravnoteži između uspješnosti klasifikatora i vremenu potrebnom da taj klasifikator nauči klasificirati primjere.

[84] Pogledati poglavlje 5.2.1, dio koji objašnjava utjecaj još jednog primjera u skupu za učenje na rezultate klasifikacije

6.4.4 Vrijeme učenja klasifikatora

Promatrajući dosada izložena mjerenja, jasno nam je kako se bolji rezultati mogu, uglavnom, uvijek dobiti povećanjem broja atributa ili broja primjera s kojima radi klasifikator. No ta povećana uspješnost ide na račun vremenske zahtjevnosti procesa učenja. Čisto teorijska razmatranja govore o polinomijalnoj vremenskoj složenosti, no zbog velikih memorijskih zahtjeva, odnosno nemogućnosti da se cijela matrica učita u memoriju, kod velikog broja primjera, ali i velikog broja atributa, može se govoriti o eksponencijalnoj vremenskoj složenosti.

6.4.4.1 Vrijeme učenja u ovisnosti o broju atributa



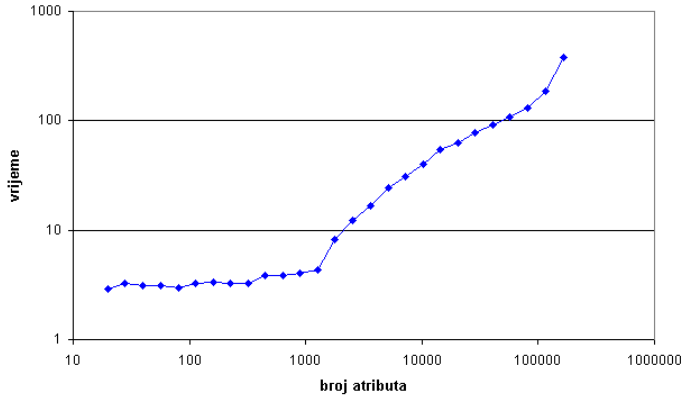
Slika 6.29 Vrijeme učenja u ovisnosti o broju atributa, odabranih na temelju IG te DF, na bazi Reuters (γ=0.01, C=1000)

Slika 6.29 prikazuje razliku u vremenima učenja klasifikatora u ovisnosti o broju atributa nad kojima uči. Prikazane su dvije krivulje. Jedna pokazuje podatke ukoliko atribute biramo na temelju mjere DF. Tada klasifikator u početku vrlo brzo uči[85], da bi nakon 1000 atributa vrijeme učenja raslo eksponencijalno s porastom broja atributa. Odabiremo li atribute na temelju IG, klasifikatoru će trebati dosta vremena kada uči i za mali broj atributa[86].

---

[85] Graf sa slike 6.26 nam govori da tada učenje i nije baš uspješno

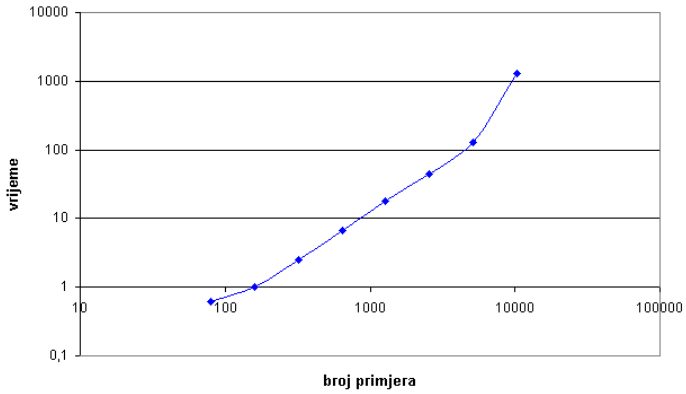
[86] Ali i za mali broj atributa, ukoliko su odabrani na temelju IG, postiže koliko-toliko dobre rezultate.



Slika 6.30 Vrijeme učenja u ovisnosti o broju atributa
(Vjesnik, DF, γ=0.01, C=1000)

Slika 6.30 pokazuje da slični rezultati vrijede i za bazu Vjesnik, nakon 1000 atributa, dolazimo u područje eksponencijalnog rasta vremena u ovisnosti o broju atributa. Ovaj nagli prijelaz u eksponencijalno područje kod gotovo iste vrijednosti broja atributa možda nam govori kako je klasifikator ušao u područje kada više ne može sve podatke, s kojima radi, zadržati u radnoj memoriji.

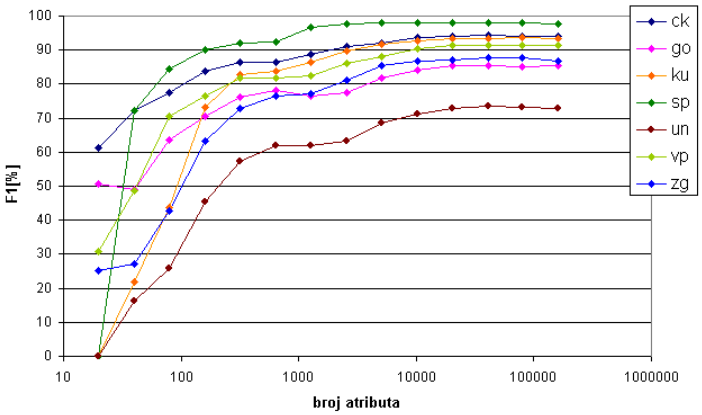
6.4.4.2 Vrijeme učenja u ovisnosti o broju primjera



Slika 6.31 Vrijeme učenja u ovisnosti o broju primjera u skupu za učenje
(Vjesnik, 10000 atributa, IG, γ=0.01, C=1000)

Rezultati sa slike 6.31 mogli su se također lako naslutiti. Jasno je vidljivo kako se radi o eksponencijalnoj funkciji što znači da je cijena povećanja broja primjera nad kojima klasifikator uči doista velika.

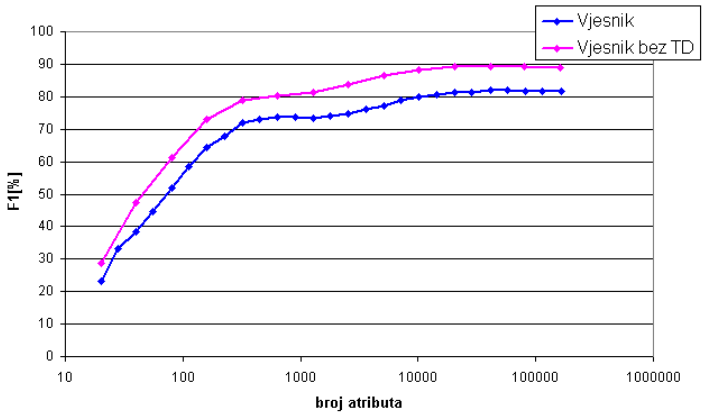
6.4.5 Uspješnost na bazi članaka Vjesnik bez kategorije TD



Slika 6.32 Ovisnost mjere F1 o broju atributa po kategorijama baze članaka Vjesnik (bez članaka iz kategorije TD, IG, γ=0.01, C=1000)

Već iz samog naziva kategorije TD (tema dana) lako se da zaključiti kako vjerojatno i nije baš najbolje definirana. Problem leži u tome što nju u tolikoj mjeri ne definira tekst, koliko kontekst. Stoga je logično kako klasifikator koji radi isključivo s tekstom neće postizati zavidne rezultate na toj kategoriji. Za pretpostaviti je kako svaki od članaka iz kategorije tema dana ima i svoju pravu kategoriju, ali je na temelju aktualnosti proglašen temom dana.

Kako bi otkrili u kolikoj mjeri ova kategorija utječe na rezultate, provedena su testiranja na bazi Vjesnik iz koje su odstranjeni svi članci s oznakom teme dana. Rezultati ovih mjerenja prikazani su na slici 6.32, a sumirani na slici 6.33.



Slika 6.33 Uspješnost izuzme li se kategorija TD iz postupka učenja i testiranja za bazu članaka Vjesnik

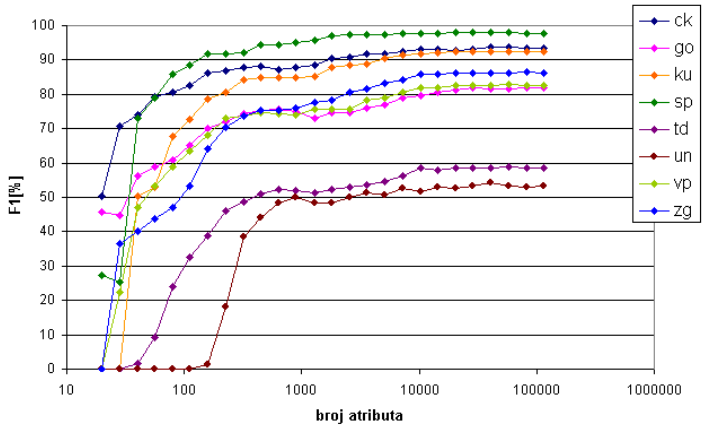
Uočava se kako su rezultati općenito bolji za oko 10%, dok su za kategoriju un (unutarnja politika) bolji i za do 25%. Iz rezultata bi se moglo zaključiti kako su temom dana najčešće proglašavani članci unutarnje politike.

6.4.6 Utjecaj apriornog znanja o tekstu na rezultate klasifikacije

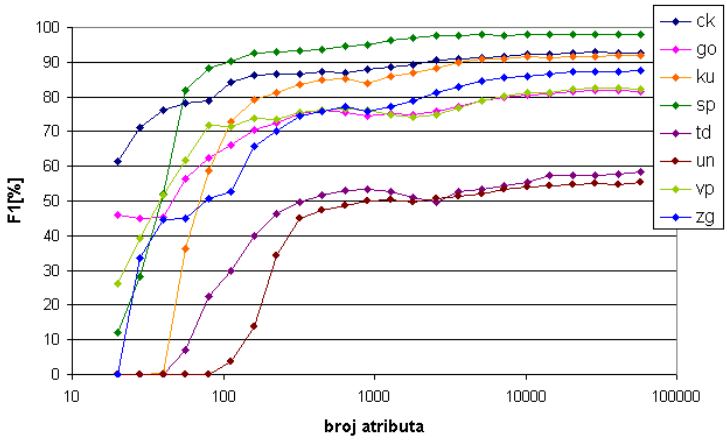
Ovaj dio mjerenja posvećen je nastojanjima da se naše predznanje o tekstu na neki način uključi u proces učenja. Prvo je razmatran utjecaj morfološkog prikaza teksta, a zatim i pretpostavka kako su riječi sadržane u naslovu i podnaslovu informativnije od drugih.

6.4.6.1 Ovisnost uspješnosti o morfološkom prikazu teksta

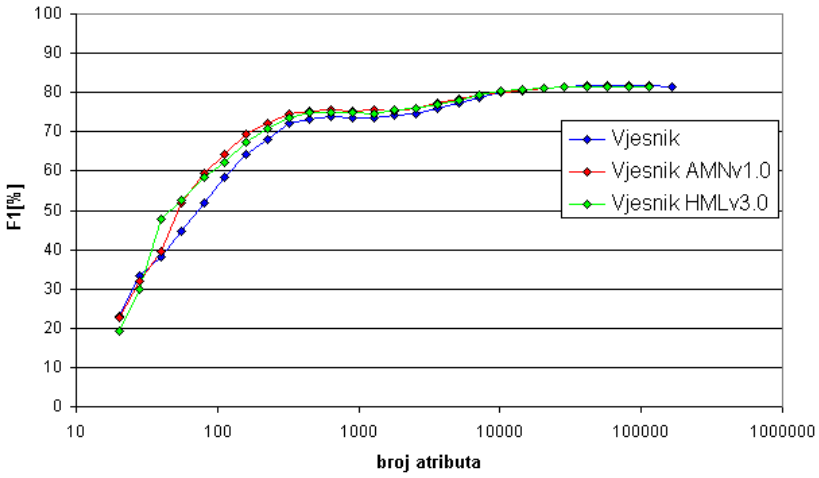
Pri radu s bazom Reuters, odnosno bilo kojim drugim tekstom na engleskom jeziku, nemamo potrebu za morfološkom obradom. Engleski je jezik morfološki vrlo siromašan. Hrvatski, naprotiv, obiluje različitim oblicima riječi, te bi se uz dobar algoritam mogla postići znatna redukcija dimenzionalnosti. Utjecaj takve redukcije na efikasnost klasifikatora prikazan je na slikama 6.34 i 6.35.



Slika 6.34 Uspješnost u ovisnosti o broju atributa na bazi članaka Vjesnik HMLv3.0 (IG, γ=0.01, C=1000)



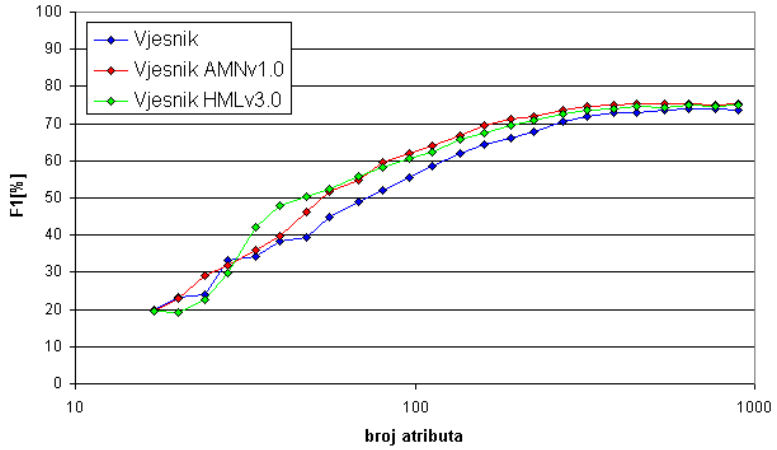
Slika 6.35 Uspješnost u ovisnosti o broju atributa na bazi članaka Vjesnik AMNv1.0 (IG, γ=0.01, C=1000)



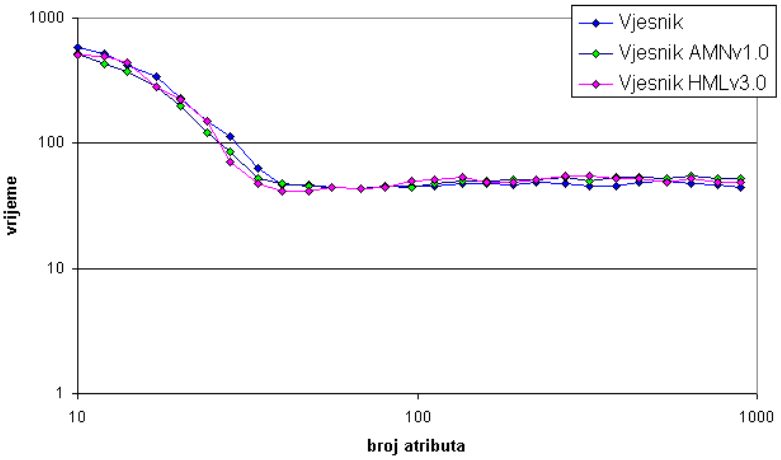
Slika 6.36 Usporedba mikrousrednjenih vrijednosti F1 za baze članaka Vjesnik, Vjesnik AMNv1.0 i Vjesnik HMLv3.0 u ovisnosti o broju atributa (IG, γ=0.01, C=1000)

Gledajući u graf sa slike 6.36 lako se može primijetiti kako su pri velikom broju atributa rezultati gotovo identični. No, taj nam dio i nije od posebnog značenja. Nama je najzanimljivije područje do 1000 atributa, jer omogućava ugodan rad s većinom algoritama strojnog učenja. Na bazi Vjesnik AMNv1.0 postižu se nešto bolji rezultati nego na Vjesnik HMLv3.0, no očito je kako obje metode pri malom broju atributa postižu zamjetno bolje rezultate od onih na bazi Vjesnik gdje nije bilo morfološke obrade.

Na slici 6.37 detaljnije je prikazan rad klasifikatora na promatranim bazama u području do 1000 atributa. Slika 6.38 prikazuje ovisnost vremena učenja o broju atributa za to isto područje. Osim za vrlo mali broj atributa, vrijeme učenja je konstantno, odnosno neovisno o broju atributa. U istom tom, konstantnom području, klasifikator dostiže maksimum uspješnosti u radu na sve tri baze. Možemo istaknuti kako je ovo optimalno područje rada klasifikatora (po pitanju broja atributa).



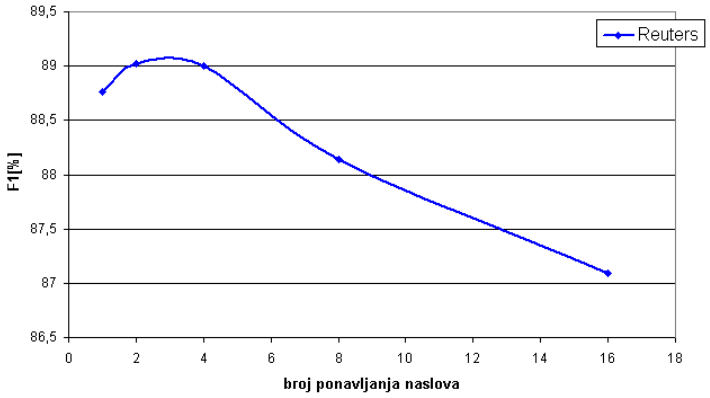
Slika 6.37 Usporedba mikrousrednjenih vrijednosti F1 za baze članaka Vjesnik, Vjesnik AMNv1.0 i Vjesnik HMLv3.0 u ovisnosti o broju atributa (IG, γ=0.01, C=1000)



Slika 6.38 Vrijeme učenja u ovisnosti o broju atributa (IG, γ=0.01, C=1000)

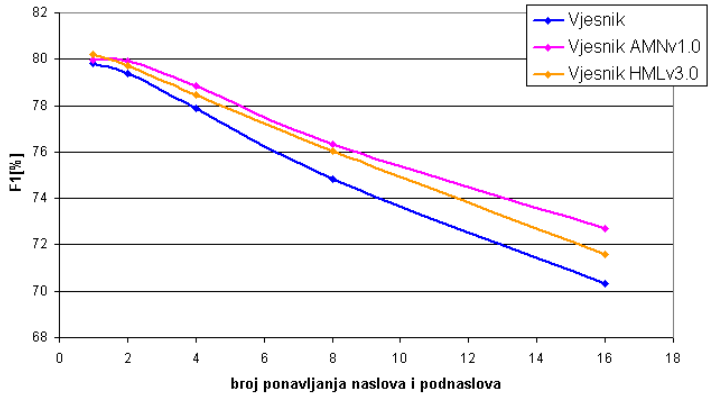
Morfološka obrada koja je rezultirala bazom Vjesnik AMNv1.0 iako dopušta pogreške, iako pojavnice ne svodi na njihovu lemu već ih zamjenjuje predstavnikom skupine, postigla je veći stupanj sažimanja koji se odrazio na bolje rezultate. Iz ovih podataka dalo bi se naslutiti kako bi se agresivnijom morfološkom obradom, onom koja bi postigla još veći stupanj sažimanja, postigli i rezultati koji bi omogućili klasifikatoru da optimalno radi već i pri 1000 atributa.

6.4.6.2 Ovisnost o broju ponavljanja naslova



Slika 6.39 Ovisnost mikrousrednjene mjere F1 o broju ponavljanja naslova članaka za bazu članaka Reuters (1000 atributa, IG, γ=0.01, C=1000)

Rezultati na bazi Reuters potvrđuju pretpostavku kako su riječi iz naslova članka informativnije od ostalih iz teksta. Prema grafu sa slike 6.39 optimalan izbor za broj ponavljanja teksta iz naslova je tri puta.



Slika 6.40 Ovisnost mikrousrednjene mjere F1 o broju ponavljanja naslova i podnaslova članaka za baze Vjesnik i Vjesnik AMNv1.0 (10000 atributa, IG, γ=0.01, C=1000)

Za razliku od rezultata na bazi Reuters, rezultati na bazama Vjesnik, Vjesnik AMNv1.0 i Vjesnik HMLv3.0 lošiji su ukoliko se naslov i podnaslov ponove još koji put, no čini se kako uz porast morfološkog sažimanja ova krivulja počinje sve više sličiti na onu sa slike 6.40

7 Zaključak

Privodeći ovaj rad kraju, prelistavajući ga u potrazi za stvarima koje sada, u ovom zaključku, treba istaknuti, čini mi se neizbježnim spomenuti razliku u živosti ovog područja godinu dana prije i sada. Najzaslužnijom osobom za to smatram svoju mentoricu prof. dr.sc. Bojanu Dalbelo-Bašić, koja je oko sebe okupila grupu mladih ljudi, uvjerila ih u važnost ovog područja, zainteresirala ih za ovu problematiku. Imao sam sreću i bio jedan od njenih prvih studenata koji je radio na ovom području.

Ne želim sada isticati rezultate ovog diplomskog na stranim bazama članaka i reći kako su rangu sa svjetskim, želim istaknuti rad na tekstu na hrvatskom jeziku. U okviru ovog diplomskog rada, svoj krajnji oblik dobila je baza novinskih članaka Vjesnik, koja će, vjerujem, sigurno još neko vrijeme biti standard za testiranje algoritama. Implementacijom SVM klasifikatora napokon je dobivena i kvalitativna mjera za neke aspekte rada Hrvatskog morfološkog leksikona prof. dr.sc. Marka Tadića kao i za proces Automatske morfološke normalizacije asistenta dipl. ing. Jana Šnajdera. Nastojao sam napraviti što više mjerenja, te ih i što jasnije objasniti, no bojim se kako su još uvijek mnoga pitanja ostala otvorena. Smatram kako je, zbog iznimnog bogatstva hrvatskog jezika, znatan napredak na ovom području moguć jedino u slučaju simbioze lingvističkog znanja i računarskog umijeća.

8 Literatura

R. Baeza-Yates, B. Ribeiro-Neto, (1999), Modern Information Retrieval, ACM Press Series/Addison Wesley, New York. [http://citeseer.ist.psu.edu/baeza-yates99modern.htm]

R. Basili, A. Moschitti, M.T. Pazienza, (2000), Language sensitive text classification, u Proceedings of RIAO '2000, Pariz, Francuska. [http://citeseer.ist.psu.edu/basili00language.html]

M.W. Berry, (2003), Survey of Text Mining: Clustering, Classification, and Retrieval, Springer Verlag Pub, New York.

D. Blair, (1992), Information retrieval and the philosophy of language, The Computer Journal, 35 (3), pp.200-207, [http://www3.oup.co.uk/computer_journal]

C. Campbell, (2000), An introduction to kernel methods, In R.J. Howlett and L.C. Jain, editors, *Radial Basis Function Networks: Design and Applications*, page 31, Springer Verlag, Berlin, [http://www.kernel-machines.org/papers/upload_13550_kmrev.ps]

N. Cristianini, J. Shawe-Taylor, (2000), An Introduction to Support Vector Machines (and other kernel based learning methods), Cambridge University Press, Cambridge.

N. Cristianini, J. Shawe-Taylor, H. Lodhi, (2001), Latent semantic kernels, Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufman. [http://citeseer.ist.psu.edu/cristianini01latent.html]

C. Cortes, V. Vapnik, (1995), Support-Vector Networks, Machine Learning, 20(3), pp.273-297, [http://citeseer.ist.psu.edu/cortes95supportvector.html]

F. Debole, F. Sebastiani, (2003), Supervised Term Weighting for Automated Text Categorization. SAC 2003, pp.784-788, [http://portal.acm.org]

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, (1990), Indexing by latent semantic analysis, Journal of the American Society for Information Science, 41(6), pp.391-407. [http://citeseer.ist.psu.edu/deerwester90indexing.html]

G. Forman, (2003), An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research 3, pp.1289-1305. [http://www.ai.mit.edu/projects/jmlr//papers/v3/forman03a.html]

M.A. Hearst, (1998), Trends and Controversies: Support Vector Machines, IEEE Intelligent Systems, 13(4). [www.computer.org/intelligent/ex1998/pdf/x4018.pdf]

A. Hotho, S. Staab, G. Stumme, (2003), WordNet improves text document clustering, Proceedings of the SIGIR 2003 Semantic Web Workshop. [http://www.aifb.uni-karlsruhe.de/WBS/aho]

S. Huang, M. O. Ward, E. A. Rundensteiner, (2003), Exploration of dimensionality reduction for text visualization, Technical Report TR-03-14, Worcester Polytechnic Institute. [http://citeseer.ist.psu.edu/huang03exploration.html]

T. Joachims, (2001), Learning to classify text using support vector machines, Kluwer, Massachusetts.

T. Joachims, (1998), Text categorization with support vector machines: Learning with many relevant features, In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pp.137-142, Berlin, Springer, [http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_98a.ps.gz]

J. Kandola, J. Shawe-Taylor, N. Cristianini, (2002), On the application of diffusion kernel to text data, Neurocolt, NeuroCOLT Technical Report NC-TR-02-122, [http://www.cg.ensmp.fr/~vert/bibli2/kernel.html]

J. Karlgren, Stylistic Experiments for Information Retrieval, (2000), Ph D Dissertation, Department of linguistics, Stockholm University, Stockholm. [http://www.sics.se/~jussi/]

G. Karypis, E.-H. Han, (2000), Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization and Retrieval, Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, pp.12-19. [http://citeseer.ist.psu.edu/article/karypis00fast.html]

G. Karypis, E.-H. Han, (2000b) Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization, Technical Report TR-00-0016, University of Minnesota. [http://citeseer.ist.psu.edu/karypis00concept.html]

T. Liu, S. Liu, Z. Chen, W.-Y. Ma, (2003), An Evaluation on Feature Selection for Text Clustering, ICML2003. [http://research.microsoft.com/~zhengc]

H. Lodhi, J. Shawe-Taylor, N. Cristianini, C. Watkins, (2000), Text classification using string kernels. *NIPS*, pp.563-569, [http://www.cg.ensmp.fr/~vert/bibli2/kernel.html]

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, (2002), Text classification using string kernels, *Journal of Machine Learning Research*, vol.2, pp.419-444, [http://www.cg.ensmp.fr/~vert/bibli2/kernel.html]

T. Masuyama, H. Nakagawa, (2004), Two Step POS Selection for SVM Based Text Categorization, Special Issue on Information Technology for Web Utilization, IEICE Transactions on Information and Systems, E87-D(2), pp.15-21. [http://www.r.dl.itc.u-tokyo.ac.jp/~nakagawa/publication-e.html]

K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, (2001), *An introduction to kernel-based learning algorithms*, IEEE Transactions on Neural Networks, 12(2), pp.181-201, [http://www.kernel-machines.org/papers/upload_24942_reviewSV_TNN01.ps.gz]

M. Porter, (1980), An algorithm for suffix stripping, Program (Automated Library and Information Systems), 14(3), pp.130-137.

M. Rogati, Y. Yang, (2002), High-performing feature selection for text classification, CIKM 2002, pp.659-661. [http://portal.acm.org/]

G. Salton, C. Buckley, (1998), Term weighting approaches in automatic text retrieval, *Information Processing and Managment*, 24(5), pp.513-523.

B. Schölkopf, (1997), *Support Vector Learning* R. Oldenbourg Verlag, Munich, [http://www.kernel-machines.org/papers/book_ref.ps.gz]

B. Schölkopf, A. J. Smola, K. Müller, (1999), Kernel principal component analysis, B. Schölkopf, C. Burges, A. Smola , *Advances in Kernel Methods - Support Vector Learning*, pp.327-352. MIT Press, [http://www.cg.ensmp.fr/~vert/bibli2/kernel.html]

B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A.J. Smola, (1999), Input space versus feature space in kernel-based methods. IEEE Transactions On Neural Networks 10(5), pp.1000-1017, [http://axiom.anu.edu.au/~smola/papers/SchMikBurKnietal99.pdf]

F. Sebastiani, (1999), A Tutorial on Automated Text Categorisation. In Analia Amandi and Ricardo Zunino, editors, Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, pp.7-35, Buenos Aires, Argentina. [http://citeseer.ist.psu.edu/sebastiani99tutorial.html]

F. Sebastiani, (2002), Machine Learning in Automated Text Categorization, ACM Computing Surveys, 34(1), pp.1–47. [www.isti.cnr.it/People/F.Sebastiani/ Publications/ACMCS02.pdf]

M. Slaney, D. Ponceleon, (2001), Hierarchical segmentation using latent semantic indexing in scale space, Proceedings of the 2001 ICASSP, Salt Lake City, Utah. [http://www.slaney.org/malcolm/pubs.html]

A. J. Smola, (1998), *Learning with Kernels*, PhD thesis, Technische Universität Berlin, [http://www.kernel-machines.org/papers/Smola98.ps.gz]

T. Strzalkowski, A. Hulth, J. Karlgren, J. Perez-Carballo, P. Tapanainen, N. Till, (1999), Natural Language Information Retrieval: TREC-8 Report', Proceedings of the 8th Text Retrieval Conference, Gaithersburg, Maryland, National Institute of Standards and Technology. [http://trec.nist.gov/pubs/trec8/papers/ge8adhoc2.pdf]

M. Tadić, (1994), *Računalna obrada morfologije hrvatskoga književnog jezika*, doktorska disertacija, Sveučilište u Zagrebu. [http://www.hnk.ffzg.hr/mt]

S. Viestam, (2001), Three methods for keyword extraction, Master's Thesis, Uppsala University. [http://stp.ling.uu.se/~matsd/thesis]

A. Vinokourov, J. Shawe-Taylor, N. Cristianini, (2002), Finding language-independent semantic representation of text using kernel canonical correlation analysis, Neurocolt, NeuroCOLT Technical Report NC-TR-02-119, [http://www.cg.ensmp.fr/~vert/bibli2/kernel.html]

K. Williams, R. Calvo, (2002), A Framework for Document Categorization, Proceedings of the 7th Australasian Document Computing Symposium. [http://www.ee.usyd.edu.au/~kenw]

Y. Yang, J.O. Pedersen, (1997), A Comparative Study on Feature Selection in Text Categorization, Proceedings of the Fourteenth International Conference on Machine Learning, pp.412-420. [http://citeseer.ist.psu.edu/yang97comparative.html]

Y. Yang, X. Liu, (1999), A re-examination of text categorization methods, SIGIR-99, [http://citeseer.ist.psu.edu/yang99reexamination.html]

9 Dodaci

9.1 Primjer rada klasifikatora teksta

Smisao ovog poglavlja je da na primjeru jednog skupa za učenje i jednog skupa za testiranje demonstrira rad najvažnijih metoda iz ovog rada. Skupovi su generirani modificiranjem naslova poglavlja triju knjiga različitih tematika. Sukladno tome i primjeri unutar skupova podijeljeni su u 3 kategorije:

1. nn - neuronske mreže
2. el - elektronika
3. dm - diskretna matematika.

U tablici 9.1 prikazan je skup za učenje, a u tablici 9.2 skup za testiranje. Oba skupa sastoje se od po 4 primjera iz svake kategorije.

nn1:	Generalizacija modela neuronskih mreža
nn2:	Umjetni neuron
nn3:	Optičke neuronske mreže
nn4:	Algoritam učenja i klasifikacija
el1:	Definicija napona praga
el2:	Kapacitet reverzno polariziranog pn-spoja
el3:	Lavinski proboj
el4:	Brzina rasta epitaksijalnog sloja
dm1:	Booleove funkcije
dm2:	Kvocjentno polje prstena polinoma
dm3:	Linearne rekurzivne relacije
dm4:	Brzina Euklidova algoritma

Tablica 9.1 Skup primjera za učenje

nn5:	Što su umjetne neuronske mreže?
nn6:	Biološki neuron
nn7:	Vrste umjetnih neuronskih mreža
nn8:	Algoritam učenja RBF mreža
el5:	Reverzno ili nepropusno polariziran pn-spoj
el6:	Brzina emiterske rekombinacije
el7:	Prikaz Fermi-Diracove funkcije
el8:	Tunelski ili Zenerov proboj
dm5:	Booleove algebre
dm6:	Definicija prstena polinoma
dm7:	Euklidov algoritam
dm8:	Nehomogene rekurzivne relacije

Tablica 9.2 Skup primjera za testiranje

9.1.1 Prikaz na nivou riječi

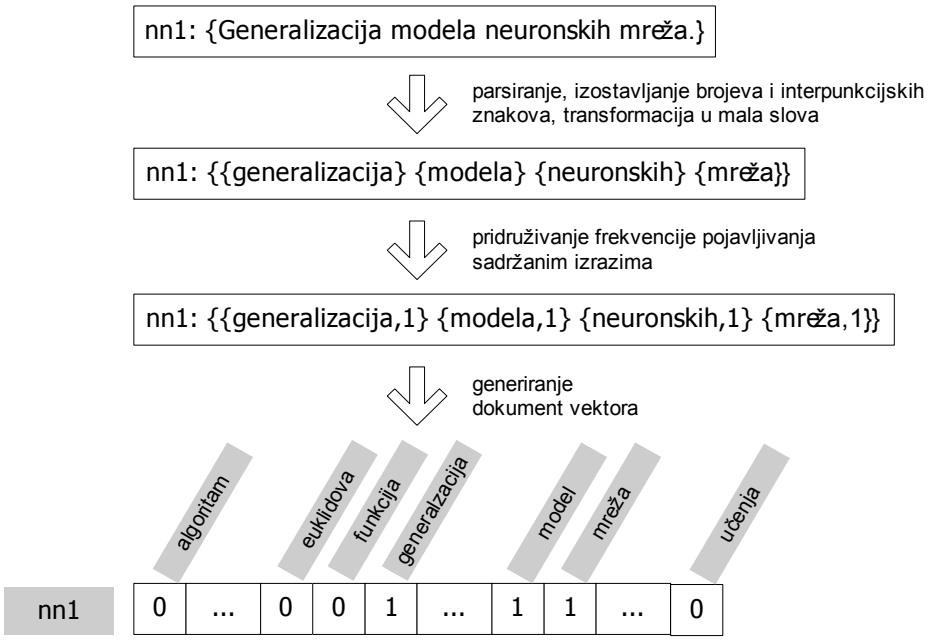
Dokumente na nivou riječi prikazujemo izraz-dokument matricom. Prvi je korak u tom postupku generiranje rječnika na temelju skupa primjera za učenje.

Parser čita tekst primjera, određuje granice izraza unutar njega; slova transformira u mala (ukoliko to već nisu), te odbacuje brojeve i interpunkcijske znakove. Prolaskom kroz sve primjere skupa za učenje generira se lista svih različitih izraza koji se pojavljuju. Ta lista tvori rječnik (tablica 9.3) na temelju kojeg se grade izraz-dokument matrice.

0:	algoritam	19:	neuron
1:	algoritma	20:	neuronske
2:	booleove	21:	neuronskih
3:	brzina	22:	optičke
4:	definicija	23:	pn
5:	epitaksijalnog	24:	polariziranog
6:	euklidova	25:	polinoma
7:	funkcije	26:	polje
8:	generalizacija	27:	praga
9:	i	28:	proboj
10:	kapacitet	29:	prstena
11:	klasifikacija	30:	rasta
12:	kvocjentno	31:	rekurzivne
13:	lavinski	32:	relacije
14:	linearne	33:	reverzno
15:	modela	34:	sloja
16:	mreža	35:	spoja
17:	mreže	36:	umjetni
18:	napona	37:	učenja

Tablica 9.3 Rječnik

Na temelju rječnika tvore se vektori koji predstavljaju primjere iz skupova. Način tvorbe vektora za dokument nn1 dan je na slici 9.1.



Slika 9.1 Proces generiranja dokument vektora za primjer nn1

Izraz-dokument matricu dobivao slaganjem dokument vektora u nju. Redci matrice označeni su izrazima iz rječnika, a stupci oznakama dokumenata. Redak matrice prikazuje frekvenciju pojavljivanja dotičnog izraza unutar skupa dokumenata, dok stupac prikazuje koji se izrazi i koliko često pojavljuju u dokumentu.

	nn1	nn2	nn3	nn4	el1	el2	el3	el4	dm1	dm2	dm3	dm4
algoritam	0	0	0	1	0	0	0	0	0	0	0	0
algoritma	0	0	0	0	0	0	0	0	0	0	0	1
booleove	0	0	0	0	0	0	0	0	1	0	0	0
brzina	0	0	0	0	0	0	0	1	0	0	0	1
...							...					
učenja	0	0	0	1	0	0	0	0	0	0	0	0

Tablica 9.4 Izraz-dokument matrica skupa za učenje

	nn5	nn6	nn7	nn8	el5	el6	el7	el8	dm5	dm6	dm7	dm8
algoritam	0	0	0	1	0	0	0	0	0	0	1	0
algoritma	0	0	0	0	0	0	0	0	0	0	0	0
booleove	0	0	0	0	0	0	0	0	1	0	0	0
brzina	0	0	0	0	0	1	0	0	0	0	0	0
...							...					
učenja	0	0	0	1	0	0	0	0	0	0	0	0

Tablica 9.5 Izraz-dokument matrica skupa za testiranje

9.1.2 Odstranjivanje stop riječi

Uklanjanje stop riječi vrši se najčešće u procesu generiranja rječnika. Riječ ne ulazi u rječnik ukoliko se nalazi u listi stop riječi. Sam proces objašnjen je detaljnije u poglavlju 6.2.2 o generiranju rječnika. U tablici 9.6 nalaze se primjeri stop riječi za hrvatski jezik.

a	mimo	zasad
ah	mimogred	zasada
aha	mljac	zasebice
aj	mnogo	zašto
aja	mnogočemu	zatim
ajme	mnogočime	zato
ako	mnogošta	zauvijek
akoli	mnom	zavazda
alaj	mnome	zazu
ali	mog	zbilja
ama	moga	zbog
amo	moguće	zbogom
amo-tamo	moj	zimus
...	...	zum

Tablica 9.6 Primjeri stop riječi

Tablica 9.7 prikazuje rječnik prije i poslije uklanjanja stop riječi. Rječnici se razlikuju samo u jednoj instanci, a to je veznik i.

0:	algoritam	19:	neuron	0:	algoritam	19:	neuronske
1:	algoritma	20:	neuronske	1:	algoritma	20:	neuronskih
2:	booleove	21:	neuronskih	2:	booleove	21:	optičke
3:	brzina	22:	optičke	3:	brzina	22:	pn
4:	definicija	23:	pn	4:	definicija	23:	polariziranog
5:	epitaksijalnog	24:	polariziranog	5:	epitaksijalnog	24:	polinoma
6:	euklidova	25:	polinoma	6:	euklidova	25:	polje
7:	funkcije	26:	polje	7:	funkcije	26:	praga
8:	generalizacija	27:	praga	8:	generalizacija	27:	proboj
9:	i	28:	proboj	9:	kapacitet	28:	prstena
10:	kapacitet	29:	prstena	10:	klasifikacija	29:	rasta
11:	klasifikacija	30:	rasta	11:	kvocjentno	30:	rekurzivne
12:	kvocjentno	31:	rekurzivne	12:	lavinski	31:	relacije
13:	lavinski	32:	relacije	13:	linearne	32:	reverzno
14:	linearne	33:	reverzno	14:	modela	33:	sloja
15:	modela	34:	sloja	15:	mreža	34:	spoja
16:	mreža	35:	spoja	16:	mreže	35:	umjetni
17:	mreže	36:	umjetni	17:	napona	36:	učenja
18:	napona	37:	učenja	18:	neuron		

Tablica 9.7 Rječnik prije (lijevo) i poslije (desno) uklanjanja stop riječi

Razlozi ovako maloj razlici leže u činjenici što je skup za učenje sastavljen od malog broja primjera te kako su primjeri zapravo naslovi koji, po svojoj prirodi, trebaju u što manje riječi nositi što više informacije, te stoga ne obiluju stop riječima.

9.1.3 Prikaz u sažetom libsvm formatu

Pogledamo li cijelu matricu skupa za učenje, vidjet ćemo da je tek oko 8% polja popunjeno. U realnim primjerima taj se broj kreće i ispod 0,1%. Ovdje je situacija nešto drugačija, jer su zbog demonstrativnih razloga birani slični primjeri. Očita je potreba za prikazom u nekoj od sažetih formi, radi uštede memorijskog prostora.

Libsvm format objašnjen je u poglavlju o generiranju matrice skupa za učenje. Svaki dokument prikazan je retkom matrice, a na početku svakog retka nalazi se oznaka pripadnosti ili nepripadnosti kategoriji.

1	8:1 14:1 15:1 20:1	1	16:1 19:1
1	18:1 35:1	1	18:1
1	16:1 19:1 21:1	1	15:1 20:1
1	0:1 10:1 36:1	1	0:1 15:1 36:1
-1	4:1 17:1 26:1	-1	22:1 32:1
-1	9:1 22:1 23:1 32:1 34:1	-1	3:1
-1	12:1 27:1	-1	7:1
-1	3:1 5:1 29:1 33:1	-1	27:1
-1	2:1 7:1	-1	2:1
-1	11:1 24:1 25:1 28:1	-1	4:1 24:1 28:1
-1	13:1 30:1 31:1	-1	0:1
-1	1:1 3:1 6:1	-1	30:1 31:1

Tablica 9.8 Prikaz matrice za učenje (lijevo) i matrice za testiranje (desno) u libsvm formatu (kategorija nn)

9.1.4 Lematizacija

Svaka riječ iz teksta zamjenjuje se lemom, tj. osnovnim oblikom te riječi. Taj posao obavlja se pomoću posebnog programa, lematizatora. Tablice 9.9 i 9.10 prikazuju lematizirane skupove primjera.

nn1:	Generalizacija model neuronski mreža
nn2:	Umjetni neuron
nn3:	Optički neuronski mreža
nn4:	Algoritam učenje i klasifikacija
el1:	Definicija napon prag
el2:	Kapacitet reverzni polarizirani pn-spoj
el3:	Lavinski proboj
el4:	Brzina rast epitaksijalni sloj
dm1:	Boole funkcija
dm2:	Kvocjentni polje prsten polinom
dm3:	Linearni rekurzivni relacija
dm4:	Brzina Euklid algoritam

Tablica 9.9 Lematizirani skup primjera za učenje

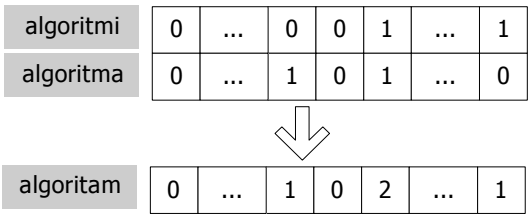
nn5:	Što biti umjetni neuronski mreža?
nn6:	Biološki neuron
nn7:	Vrsta umjetni neuronski mreža
nn8:	Algoritam učenje RBF mreža
el5:	Reverzni ili nepropusni polarizirani pn-spoj
el6:	Brzina emiterski rekombinacija
el7:	Prikaz Fermi-Dirac funkcija
el8:	Tunelski ili Zener proboj
dm5:	Boole algebra
dm6:	Definicija prsten polinom
dm7:	Euklid algoritam
dm8:	Nehomogeni rekurzivni relacija

Tablica 9.10 Lematizirani skup primjera za testiranje

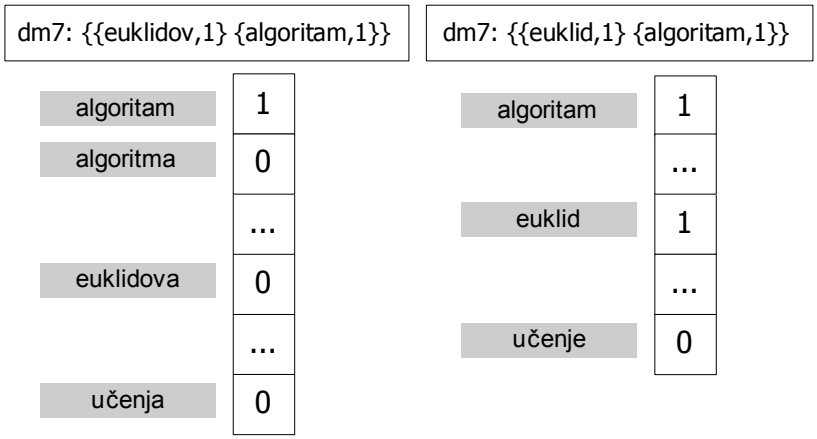
Na temelju lematiziranog skupa primjera za učenje, postupcima kao u poglavljima 9.1 i 9.2, dobivamo rječnik iz tablice 9.11. Iz ovog rječnika i lematiziranih skupova grade se matrice za učenje i testiranje. Dok je u matrici za učenje došlo samo do zbrajanja nekih redaka, primjer je na slici 9.2, kod skupa za testiranje sada su obuhvaćene i neke riječi koje su prije bile odbačene, jer takav oblik riječi nije postojao u rječniku. Na primjeru sa slike 9.3, vidi se kako je za nelematizirani slučaj (lijevi dio slike), pri generiranju dokument vektora, riječ euklidov jednostavno zanemarena, dok je kod lematiziranih primjera (desni dio slike) iskorištena u konstrukciji dokument vektora.

0:	algoritam	17:	neuronski
1:	boole	18:	optički
2:	brzina	19:	pn
3:	definicija	20:	polarizirani
4:	epitaksijalni	21:	polinom
5:	euklid	22:	polje
6:	funkcija	23:	prag
7:	generalizacija	24:	proboj
8:	kapacitet	25:	prsten
9:	klasifikacija	26:	rast
10:	kvocjentni	27:	rekurzivni
11:	lavinski	28:	relacija
12:	linearni	29:	reverzni
13:	model	30:	sloj
14:	mreža	31:	spoj
15:	napon	32:	umjetni
16:	neuron	33:	učenje

Tablica 9.11 Rječnik lematiziranog skupa za učenje



Slika 9.2 Lematizacija na matricu za učenje



Slika 9.3 Utjecaj lematizacije na matricu za testiranje

1	7:1 13:1 14:1 17:1	1	14:1 17:1 32:1
1	16:1 32:1	1	16:1
1	14:1 17:1 18:1	1	14:1 17:1 32:1
1	0:1 9:1 33:1	1	0:1 14:1 33:1
-1	3:1 15:1 23:1	-1	19:1 20:1 29:1 31:1
-1	8:1 19:1 20:1 29:1 31:1	-1	2:1
-1	11:1 24:1	-1	6:1
-1	2:1 4:1 26:1 30:1	-1	24:1
-1	1:1 6:1	-1	1:1
-1	10:1 21:1 22:1 25:1	-1	3:1 21:1 25:1
-1	12:1 27:1 28:1	-1	0:1 5:1
-1	0:1 2:1 5:1	-1	27:1 28:1

Tablica 9.12 Prikaz matrice lematiziranog skupa za učenje (lijevo) i martrice lematiziranog skupa za testiranje (desno) u sažetom libsvm formatu (kategorija nn)

9.1.5 Odabir atributa na temelju informacijske dobiti

Informacijska dobit zadana je izrazom 3.3; to je općenita formula, a njen oblik koji odgovara našem binarnom klasifikatoru dan je izrazom 9.1.

$$\begin{aligned} IG(t) = & -p(c_1)\cdot\log(p(c_1)) - p(c_{-1})\cdot\log(p(c_{-1})) \\ & + p(t)\cdot\left[p(c_1\mid t)\cdot\log(p(c_1\mid t)) + p(c_{-1}\mid t)\cdot\log(p(c_{-1}\mid t))\right] \\ & + p(\neg t)\cdot\left[p(c_1\mid \neg t)\cdot\log(p(c_1\mid \neg t)) + p(c_{-1}\mid \neg t)\cdot\log(p(c_{-1}\mid \neg t))\right] \end{aligned} \quad (9.1)$$

Značenja oznaka su sljedeća: oznaka t predstavlja prisustvo promatranog izraza, dok oznaka ¬t predstavlja odsustvo tog izraza, oznaka c1 označava pripadnost kategoriji 1, dok c-1 označava pripadnost kategoriji -1. Pojedinosti oko izračunavanja pojedinih komponenti formule dane su sljedećim izrazima:

$$p(c_1) = \frac{broj\_dokumenata\_iz\_kategorije\_1}{ukupan\_broj\_dokumenata} \quad (9.2)$$

$$p(c_{-1}) = \frac{broj\_dokumenata\_iz\_kategorije\_-1}{ukupan\_broj\_dokumenata} \quad (9.3)$$

$$p(t) = \frac{broj\_dokumenata\_koji\_sadrže\_izraz\_t}{ukupan\_broj\_dokumenata} \quad (9.4)$$

$$p(\neg t) = \frac{broj\_dokumenata\_koji\_ne\_sadrže\_izraz\_t}{ukupan\_broj\_dokumenata} \quad (9.5)$$

$$p(c_1\mid t) = \frac{broj\_dokumenata\_iz\_1\_koji\_sadrže\_izraz\_t}{broj\_dokumenata\_iz\_kategorije\_1} \quad (9.6)$$

$$p(c_{-1}\mid t) = \frac{broj\_dokumenata\_iz\_-1\_koji\_sadrže\_izraz\_t}{broj\_dokumenata\_iz\_kategorije\_-1} \quad (9.7)$$

$$p(c_1\mid \neg t) = \frac{broj\_dokumenata\_iz\_1\_koji\_ne\_sadrže\_izraz\_t}{broj\_dokumenata\_iz\_kategorije\_1} \quad (9.8)$$

$$p(c_{-1}\mid \neg t) = \frac{broj\_dokumenata\_iz\_-1\_koji\_ne\_sadrže\_izraz\_t}{broj\_dokumenata\_iz\_kategorije\_-1} \quad (9.9)$$

Izračunamo li informacijsku dobit za sve izraze iz rječnika dobit ćemo ovakve redoslijede, ovisno o kategoriji koju pokušavamo naučiti.

0:	mreža	brzina	algoritam
1:	neuronski	definicija	brzina
2:	algoritam	napon	boole
3:	klasifikacija	prag	euklid
4:	optički	kapacitet	funkcija
5:	učenje	pn	kvocjentni
6:	neuron	polarizirani	linearni
7:	umjetni	reverzni	polinom
8:	generalizacija	spoj	polje
9:	model	epitaksijalni	prsten
10:	brzina	lavinski	rekurzivni
11:	definicija	proboj	relacija
12:	boole	rast	mreža
13:	epitaksijalni	sloj	neuronski
14:	funkcija	mreža	klasifikacija
15:	euklid	neuronski	optički

Tablica 9.13 Redoslijed najvrjednijih 16 atributa prema informacijskoj dobiti, za kategorije: nn (lijevo), el (sredina), dm (desno)

Tablica 9.13 prikazuje redoslijed 16 najvrjednijih atributa prema informacijskoj dobiti po kategorijama. Razlog zašto imamo 3 stupca leži u činjenici kako imamo 3 binarna klasifikatora, a svaki od njih na drugačiji način pridjeljuje primjerima oznake kategorija.

0:	algoritam
1:	boole
2:	brzina
3:	definicija
4:	epitaksijalni
5:	euklid
6:	funkcija
7:	generalizacija
8:	klasifikacija
9:	model
10:	mreža
11:	neuron
12:	neuronski
13:	optički
14:	umjetni
15:	učenje

Tablica 9.14 Rječnik za nn kategoriju

Od 34 moguća atributa odabrano je najboljih 16 (tablica 9.13 lijevo) te su poslagani u rječnik (tablica 9.14). Na temelju ovih atributa izgrađene su matrice za učenje i testiranje (tablica 9.15).

1	7:1 9:1 10:1 12:1	1	10:1 12:1 14:1
1	11:1 14:1	1	11:1
1	10:1 12:1 13:1	1	10:1 12:1 14:1
1	0:1 8:1 15:1	1	0:1 10:1
-1	3:1	-1	
-1		-1	2:1
-1		-1	6:1
-1	2:1 4:1	-1	
-1	1:1 6:1	-1	1:1
-1		-1	3:1
-1		-1	0:1 5:1
-1	0:1 2:1 5:1	-1	

Tablica 9.15 Prikaz matrice za učenje (lijevo) i matrice za testiranje (desno) nakon odabira atributa (kategorija nn)

Na temelju tablice 9.15 nije nelogično zaključiti kako algoritam bira atribute tako da dobro opišu pozitivnu kategoriju primjera, jer neki od negativnih primjera nisu opisani niti jednim atributom.

9.1.6 TFIDF prikaz

Prikaz koji dodjeljujemo izrazu sastoji se od 3 komponente: komponente dokumenta (izraz 9.10), komponente skupa (izraz 9.11) predstavljene inverznom frekvencijom dokumenata, te euklidske normalizacijske komponente (izraz 9.12).

$$TF(w_i,d) \tag{9.10}$$

$$\log\frac{|D|}{DF(w_i)} \tag{9.11}$$

$$\frac{1}{\sqrt{\sum x_j^2}} \tag{9.12}$$

Prikaz koji nastaje kombinacijom ovih komponenti dan je izrazom 9.13.

$$x_i = \frac{TF(w_i,d)\cdot\log\left(\frac{|D|}{DF(w_i)}\right)}{\sqrt{\sum_j\left[TF(w_j,d)\cdot\log\left(\frac{|D|}{DF(w_i)}\right)\right]^2}} \tag{9.13}$$

Formule za izračunavanje pojedinih komponenti ovog prikaza dane su sljedećim izrazima.

$$TF(w_i, d) = broj\_pojavljivanja\_izraza\_w_i\_u\_dokumentu\_d \tag{9.14}$$

$$|D| = broj\_dokumenata \tag{9.15}$$

$$DF(w_i) = broj\_dokumenata\_u\_kojima\_se\_pojavljuje\_w_i \tag{9.16}$$

Treba naglasiti kako se izrazi 9.14 i 9.15 računaju samo na skupu za učenje, a zatim se prenose i na skup za testiranje. Razlog tomu je što skup za testiranje tretiramo kao niz nezavisnih primjera, dok se prema skupu za učenje odnosimo kao prema cjelini koja nosi određenu informaciju. Primjer preslikavanja u tfidf prikaz dan je na slici 9.4.

$$\underbrace{\begin{pmatrix}1\\0\\0\\2\\1\\0\\3\\0\end{pmatrix}}_{\text{dokument vektor tf}} \Rightarrow \underbrace{\begin{pmatrix}3.2&0&0&0&0&0&0&0\\0&5.0&0&0&0&0&0&0\\0&0&1.5&0&0&0&0&0\\0&0&0&2.4&0&0&0&0\\0&0&0&0&1.9&0&0&0\\0&0&0&0&0&9.5&0&0\\0&0&0&0&0&0&11.0&0\\0&0&0&0&0&0&0&3.1\end{pmatrix}}_{\text{komponenta kolekcije idf}} \cdot \begin{pmatrix}1\\0\\0\\2\\1\\0\\3\\0\end{pmatrix} = \underbrace{\begin{pmatrix}3.2\\0\\0\\4.8\\1.9\\0\\33.0\\0\end{pmatrix}}_{\text{tf-idf prikaz}} \Rightarrow \underbrace{\begin{pmatrix}0.095\\0\\0\\0.143\\0.057\\0\\0.983\\0\end{pmatrix}}_{\text{normirani tf-idf prikaz}}$$

Slika 9.4 Primjer preslikavanja u tfidf prikaz

Prevedemo li sada matrice za učenje i testiranje u ovaj novi oblik reprezentacije, dobit ćemo matrice iz tablice 9.16.

1	7:0.5735 9:0.5735 10:0.4135
1	12:0.4135
1	11:0.7071 14:0.7071
1	10:0.5048 12:0.5048 13:0.7001
-1	0:0.4542 8:0.6299 15:0.6299
-1	3:1.0000
-1	
-1	
-1	2:0.5848 4:0.8111
-1	1:0.7071 6:0.7071
-1	
-1	
	0:0.5048 2:0.5048 5:0.7001

1	10:0.5048 12:0.5048 14:0.7001
1	11:1.0000
1	10:0.5048 12:0.5048 14:0.7001
1	0:0.7071 10:0.7071
-1	
-1	2:1.0000
-1	6:1.0000
-1	
-1	1:1.0000
-1	3:1.0000
-1	0:0.5848 5:0.8111
-1	

Tablica 9.16 Matrica za učenje (lijevo), matrica za testiranje (desno). (kategorija nn)

9.1.7 Učenje i klasifikacija

Kroz proces učenja program određuje parametre α i b izraza (5.29). Rezultat učenja za nn klasifikator su podaci u tablici 9.17.

```
1  svm_type c_svc
2  kernel_type rbf
3  gamma 0.01
4  nr_class 2
5  total_sv 8
6  rho -2.1477
7  label 1 –1
8  nr_sv 4 4
9  SV
10 70.23817350302504 7:0.573554 9:0.573554 10:0.413565 12:0.413565
11 99.39089310277026 11:0.707107 14:0.707107
12 70.23117638248195 10:0.50486 12:0.50486 13:0.700166
13 105.2377135855445 0:0.45423 8:0.62995 15:0.62995
14 -1.621707964574358 3:1
15 -316.2977698416948
16 -1.621391758975979 1:0.707107 6:0.707107
17 -25.55708700857668 0:0.50486 2:0.50486 5:0.700166
```

Tablica 9.17 Izlazni podaci procesa učenja

Od prikazanih podataka valja izdvojiti redak 3 u kojem se nalazi parametar rbf jezgrene funkcije gama, te retke 10 do 17, gdje je prvo zapisan parametar α, a zatim i potporni vektor na kojeg se taj parametar odnosi.

Zanimljivo je primijetiti kako je klasifikator učeći na 12 primjera njih 8 odabrao kao potporne vektore. U tih 8 ulaze sva četiri iz kategorije nn (oznaka 1) te još 4 iz kategorije ne pripada nn (oznaka -1). Razlog ovako velikom broju potpornih vektora je različitost dokumenata unutar skupa za učenje.

Uvrštavanjem podataka o potpornim vektorima i podataka o jezgrenoj funkciji u izraz (9.17) dobivamo funkciju odlučivanja koja daje rezultate iz tablice 9.18.

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{9.17}$$

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \tag{9.18}$$

dokument	klasifikacija
nn5	1
nn6	1
nn7	1
nn8	1
el5	-1
el6	-1
el7	-1
el8	-1
dm5	-1
dm6	-1
dm7	-1
dm8	-1

Tablica 9.18 Rezultati klasifikacije (kategorija nn)

9.1.8 Vrednovanje rezultata

Izvođenjem postupka koji je sustavno opisivan za nn kategoriju i za el i dm kategorije dobit će se rezultati iz tablice 9.19.

	nn		el		dm	
dokument	klasifikacija	ispravno	klasifikacija	ispravno	klasifikacija	ispravno
nn5	1	1	-1	-1	-1	-1
nn6	1	1	-1	-1	-1	-1
nn7	1	1	-1	-1	-1	-1
nn8	1	1	-1	-1	-1	-1
el5	-1	-1	1	1	-1	-1
el6	-1	-1	-1	1	1	-1
el7	-1	-1	-1	1	1	-1
el8	-1	-1	1	1	-1	-1
dm5	-1	-1	-1	-1	1	1
dm6	-1	-1	1	-1	1	1
dm7	-1	-1	-1	-1	1	1
dm8	-1	-1	-1	-1	1	1

Tablica 9.19 Rezultati klasifikacije

Uspoređujući izlaze klasifikatora sa stvarnim vrijednostima popunjavamo tablicu 9.20. Oznake su objašnjenje u poglavlju o mjerama uspješnosti.

	TP	NP	NN	TN
nn	4	0	0	8
el	2	1	2	7
dm	4	2	0	6

Tablica 9.20 Tablica mogućih ishoda

Imamo li podatke iz tablice mogućih ishoda, lako ćemo prema izrazima (4.7), (4.9) i (4.11) izračunati odziv, preciznost i F_1 mjeru.

	odziv	preciznost	F_1
nn	1	1	1
el	0,5	0,666667	0,571429
dm	1	0,666667	0,8

Tablica 9.21 Odziv, preciznost i F_1

Rezultati u tablici 9.21 govore nam o uspješnosti klasifikatora pri klasifikaciji pojedine kategorije. Želimo li imati vrijednost koja će nam predstavljati uspješnost klasifikatora na svim kategorijama, koristimo se metodom mikrousrednjavanja (izraz 4.13). Kako bi dobili vrijednost za mikrousrednjenu F_1 mjera, moramo napraviti novu

tablicu mogućih ishoda koja će sadržavati prosječne vrijednosti po pojedinim stavkama (tablica 9.22).

	TP^{avg}	NP^{avg}	NN^{avg}	TN^{avg}
mikro	3,333333	1	0,666667	7

Tablica 9.22 Mikrousrednjena tablica mogućih ishoda

Na temelju ove tablice lako se računa mikrousrednjena F_1 mjera koja iznosi: 0,790476.