Klasifikacija tekstualnih dokumenata

Bukvić, Kristina

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Kristina Bukvić

KLASIFIKACIJA TEKSTUALNIH DOKUMENATA

Diplomski rad

Voditelj rada:
doc.dr.sc. Pavle Goldstein

Zagreb, rujan, 2021.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Strojno učenje je grana umjetne inteligencije koja se bavi oblikovanjem algoritama čiji je cilj procjena budućeg ponašanja uzoraka prema utvrđenom kriteriju točnosti. Tehnike strojnog učenja mogu se podijeliti u dvije kategorije: nadzirano učenje *(eng. supervised learning)* i nenadzirano učenje *(eng. unsupervised learning)*. Nadzirano učenje za izgradnju modela koristi podatke za koje su unaprijed poznati razredi, $(x, y) = (ulaz, izlaz)$, pri čemu je x ulazna vrijednost, a y ciljana vrijednost. Prilikom učenja, algoritam pronalazi funkciju $\hat{y} = f(x)$ koja za nove ulazne varijable x predviđa izlazne varijable $\hat{y}$. Jedno od područja u kojem se koristi nadzirano učenje je klasifikacija dokumenata.

Ubrzanim rastom dostupnih podataka i informacija, dolazi do razvoja tehnika za klasifikaciju dokumenata. One omogućavaju organizaciju i upravljanje tekstualnim podacima. Prema tome, glavni zadatak klasifikacije je pojedinom tekstu ili dokumentu odrediti klasu, pri čemu je klasa skup objekata s istim karakteristikama. U ovom diplomskom radu ćemo obraditi neke od pristupa klasifikaciji teksta, a to su: model *"Bag of words"* i model *"Bag of n-grams"*.

Cilj rada je analizirati i usporediti točnost klasifikacije teksta temeljene na gore navedenim modelima. Također ćemo u modelu *"Bag of n-grams"* usporediti rezultate dobivene s obzirom na različite duljine *n-grama*. Algoritam klasifikacije koji koristimo je stroj potpornih vektora.

U ovom radu je sadržano pet poglavlja. Prvo poglavlje sadrži matematičke pojmove iz linearne algebre i optimizacije, koje ćemo koristiti u nastavku rada. Drugo poglavlje definira klasifikaciju teksta te opisuje neke modele klasifikacije. U trećem poglavlju dobivamo uvid u matematičku podlogu stroja potpornih vektora. Četvrto poglavlje opisuje primjenu modela *"Bag of n-grams"* na skupu podataka kojim se bavimo te opisuje rezultate klasifikacije dobivene klasifikatorom stroja potpornih vektora. Na kraju, u petom poglavlju opisujemo implementaciju klasifikacije dokumenata u programskom jeziku Python.

Poglavlje 1

Matematički pojmovi

Zbog razumijevanja idućih poglavlja, potrebno je uvesti neke osnovne matematičke pojmove iz linearne algebre i optimizacije. Pojmovi iz linearne algebre su iz izvora [1], a pojmovi iz optimizacije iz izvora [4].

1.1 Linearna algebra

Definicija 1.1.1. *Neka je* $\mathbb{F}$ *neki skup na kojem su definirane operacije zbrajanja*

$$+ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

i množenja

$$\cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

koje imaju iduća svojstva:

1. $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$;
2. $\exists\, 0 \in \mathbb{F}$ *sa svojstvom* $\alpha + 0 = 0 + \alpha = \alpha, \forall \alpha \in \mathbb{F}$;
3. $\forall\, \alpha \in \mathbb{F}, \exists - \alpha \in \mathbb{F}$ *tako da je* $\alpha + (-\alpha) = (-\alpha) + \alpha = 0$;
4. $\alpha + \beta = \beta + \alpha$, $\forall \alpha, \beta \in \mathbb{F}$;
5. $(\alpha\beta)\gamma = \alpha(\beta\gamma), \forall \alpha, \beta, \gamma \in \mathbb{F}$;
6. $\exists\, 1 \in \mathbb{F} \setminus \{0\}$ *sa svojstvom* $1 \cdot \alpha = \alpha \cdot 1 = \alpha$, $\forall \alpha \in \mathbb{F}$;

7. $\forall \alpha \in \mathbb{F},\ \alpha \neq 0, \exists \alpha^{-1} \in \mathbb{F}$ *tako da je* $\alpha\alpha^{-1} = \alpha^{-1}\alpha = 1$*;*

8. $\alpha\beta = \beta\alpha,\ \forall \alpha, \beta \in \mathbb{F}$*;*

9. $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma,\ \forall \alpha, \beta, \gamma \in \mathbb{F}$*.*

Tada kažemo da je $\mathbb{F}$ *polje, a elemente polja nazivamo skalarima.*

Napomena 1.1.2. *Skup realnih brojeva s uobičajenim operacijama zbrajanja i množenja je polje.*

Definicija 1.1.3. *Neka je V neprazan skup na kojem su zadane binarne operacije zbrajanja*

$$+ : V \times V \to V$$

i operacija množenja skalarima iz polja $\mathbb{F}$*,*

$$\cdot : \mathbb{F} \times V \to V.$$

Kažemo da je uređena trojka $(V, +, \cdot)$ *vektorski prostor nad poljem* $\mathbb{F}$ *ako vrijedi:*

1. $a + (b + c) = (a + b) + c,\ \forall a, b, c \in V$*;*

2. $\exists\, 0 \in V$ *sa svojstvom* $a + 0 = 0 + a = a, \forall a \in V$*;*

3. $\forall a \in V, \exists -a \in V$ *tako da je* $a + (-a) = (-a) + a = 0$*;*

4. $a + b = b + a, \forall a, b \in V$*;*

5. $\alpha(\beta a) = (\alpha\beta)a, \forall \alpha, \beta \in \mathbb{F}, \forall a \in V$*;*

6. $(\alpha + \beta)a = \alpha a + \beta a, \forall \alpha, \beta \in \mathbb{F}, \forall a \in V$*;*

7. $\alpha(a + b) = \alpha a + \alpha b, \forall \alpha \in \mathbb{F}, \forall a, b \in V$*;*

8. $1 \cdot a = a \cdot 1, \forall a \in V.$

Elemente vektorskog prostora nazivamo vektori.

Vektorski prostori nad poljem $\mathbb{R}$ nazivaju se realni vektorski prostori, a za one nad poljem $\mathbb{C}$ kažemo da su kompleksni vektorski prostori.

Neka je $n \in \mathbb{N}$, te neka $\mathbb{R}^n$ označava skup svih uređenih n-torki realnih brojeva (drugim riječima, $\mathbb{R}^n$ je Kartezijev produkt od n kopija skupa $\mathbb{R}$). Definirajmo

$$(a_1, a_2, ..., a_n) + (b_1, b_2, ..., b_n) = (a_1 + b_1, a_2 + b_2, ..., a_n + b_n)$$

i za $\alpha \in \mathbb{R}$

$$\alpha(a_1, a_2, ..., a_n) = (\alpha a_1, \alpha a_2, ..., \alpha a_n)$$

Jasno je da uz ovako definirane operacije $\mathbb{R}^n$ realan vektorski prostor.

Definicija 1.1.4. *Neka je V vektorski prostor nad poljem $\mathbb{F}$. Izraz oblika*

$$\alpha_1 a_1 + \alpha_2 a_2 + ... + \alpha_k a_k$$

pri čemu je $a_1, a_2, ..., a_k \in V$, $\alpha_1, \alpha_2, ..., \alpha_k \in \mathbb{F}$ i $k \in \mathbb{N}$, naziva se linearna kombinacija vektora $a_1, a_2, ..., a_k$ s koeficijentima $\alpha_1, \alpha_2, ..., \alpha_k$.

Definicija 1.1.5. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i*

$$S = \{a_1, a_2, ..., a_k\}, k \in \mathbb{N}$$

konačan skup vektora iz V. Ako vrijedi

$$\forall \alpha_1, \alpha_2, ..., \alpha_k \in \mathbb{F}, \sum_{i=1}^{k} \alpha_i a_i = 0 \Rightarrow \alpha_1 = \alpha_2 = ... = \alpha_k = 0,$$

kažemo da je skup S linearno nezavisan. U suprotnom kažemo da je skup S linearno zavisan.

Definicija 1.1.6. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i $S \subseteq V, S \neq \emptyset$. Linearna ljuska skupa S označava se simbolom $[S]$ i definira kao*

$$[S] = \{\sum_{i=1}^{k} \alpha_i a_i : \alpha_i \in \mathbb{F}, a_i \in S, k \in \mathbb{N}\}.$$

Dodatno, definira se $[\emptyset] = 0$.

Linearna ljuska nepraznog skupa S je, dakle, skup svih linearnih kombinacija elemenata skupa S.

Definicija 1.1.7. *Neka je V vektorski prostor i $S \subseteq V$. Kaže se da je S sustav izvodnica za V (ili da S generira V) ako vrijedi $[S] = V$.*

Definicija 1.1.8. *Konačni skup $B = \{b_1, b_2, ..., b_n\}, n \in \mathbb{N}$, u vektorskom prostoru V, naziva se baza za V ako je B linearno nezavisan sustav izvodnica za V.*

Sljedeći teorem je fundamentalan rezultat linearne algebre. Smisao je u tome da svaki vektor danog prostora možemo na jedinstven način predočiti kao linearnu kombinaciju vektora baze. Na ovaj se način svaki problem i svaki račun u tom prostoru može svesti na operiranje s konačno mnogo vektora.

Teorem 1.1.9. *Neka je V vektorski prostor nad poljem $\mathbb{F}$, te neka je $B = \{b_1, b_2, ..., b_n\}, n \in \mathbb{N}$, baza za V. Tada za svaki vektor $v \in V$ postoje jedinstveno određeni skalari $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{F}$ takvi da vrijedi $v = \sum_{i=1}^{n} \alpha_i b_i$*

Definicija 1.1.10. *Za prirodne brojeve m i n, preslikavanje*

$$A : \{1, 2, ..., m\} \times \{1, 2, ..., n\} \to \mathbb{F}$$

se naziva matrica tipa (m, n) s koeficijentima iz polja $\mathbb{F}$.

Napomena 1.1.11. *Djelovanje svake takve funkcije A piše se tablično, u m redaka i n stupaca gdje se u i-ti i j-ti stupac piše funkcijska vrijednost $A(i, j)$. U tom smislu kažemo da je A matrica s m redaka i n stupaca. Običaj je da se ta funkcijska vrijednost $A(i, j)$ označava kao a_{ij}.*

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

Definicija 1.1.12. *Neka je V vektorski prostor nad poljem $\mathbb{F}$. Skalarni produkt na V je preslikavanje:*

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$$

koje ima sljedeća svojstva:

1. $\langle x, x \rangle \geq 0, \forall x \in V;$

2. $\langle x, x \rangle = 0 \Leftrightarrow x = 0, \forall x \in V;$

3. $\langle x_1 + x_2, y \rangle = \langle x_1, y \rangle + \langle x_2, y \rangle, \forall x_1, x_2, y \in V;$

4. $\langle \alpha x, y\rangle = \alpha\langle x, y\rangle$, $\forall \alpha \in \mathbb{F}, \forall x, y \in V$;

5. $\langle x, y\rangle = \overline{\langle y, x\rangle}$, $\forall x, y \in V$.

Treba primijetiti da skalarni produkt poprima vrijednosti u polju nad kojim je dani vektorski prostor izgrađen; ako je, dakle, prostor kompleksan, zadnje svojstvo kaže da su skalarni umnošci $\langle x, y\rangle$ i $\langle y, x\rangle$ međusobno konjugirano kompleksni brojevi. Ako je pak prostor realan, skalarni umnožak bilo koja dva vektora je realan broj pa kompleksno konjugiranje nema efekta i ovo svojstvo u realnim prostorima glasi:

$$\langle x, y\rangle = \langle y, x\rangle$$

Stoga se u realnim prostorima svojstvo (5) naziva simetričnost, a u kompleksnim prostorima hermitska simetričnost. Nas će u ovom radu zanimati realan prostor.

Definicija 1.1.13. *Neka je V vektorski prostor nad poljem* $\mathbb{F}$ *s definiranim skalarnim produktom. Tada V nazivamo unitarnim prostorom.*

Definicija 1.1.14. *Euklidski prostor je unitaran realni prostor.*

Napomena 1.1.15. *U* $\mathbb{R}^n$ *skalarni produkt je obično definiran s*

$$\langle (\ x_1, ...,\ x_n), (\ y_1, ...,\ y_n)\rangle = \sum_{i=1}^{n} x_i \overline{y_i}$$

Definicija 1.1.16. *Neka je V vektorski prostor nad* $\mathbb{F}$ *i* $M \subseteq V, M \neq \emptyset$. *Ako je i* $(M, +, \cdot)$ *vektorski prostor nad* $\mathbb{F}$ *uz iste operacije iz V, kažemo da je M potprostor od V.*

Definicija 1.1.17. *Neka je V unitaran prostor. Norma na V je funkcija*

$$\|\cdot\| : V \to \mathbb{R}$$

definirana s

$$\|x\| = \sqrt{\langle x, x\rangle}.$$

Propozicija 1.1.18. *Norma na unitarnom prostoru V ima sljedeća svojstva:*

1. $\|x\| \geq 0, \forall x \in V$;

2. $\|x\| = 0 \Leftrightarrow x = 0, \forall x \in V$;

3. $\|\alpha x\| = |\alpha|\|x\|, \forall \alpha \in \mathbb{F}, \forall x \in V$;

4. $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in V$.

Definicija 1.1.19. *Neka je V unitaran prostor. Kaže se da je vektor $x \in V$ normiran ako je $||x|| = 1$.*

Neka je dan vektor $x = (x_1, x_2, ..., x_n) \in V$ nad poljem $\mathbb{R}^n$.

- 1-norma je dana sa $||x||_1 = \sum_{i=1}^{n} |x_i|$.
- Euklidska norma ili 2-norma je dana sa $||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$.
- Max-norma je dana sa $||x||_{max} = max\{|x_1|, ..., |x_n|\}$.

Definicija 1.1.20. *Svaka funkcija $|| \cdot || : V \to \mathbb{R}$ na vektorskom prostoru V sa svojstvima iz 1.1.18 naziva se norma. Tada $(V, || \cdot ||)$ zovemo normirani prostor.*

Definicija 1.1.21. *Neka je V unitaran prostor. Metrika ili udaljenost vektora x i y je funkcija*

$$d : V \times V \to \mathbb{R}$$

definirana s

$$d(x, y) = ||x - y||.$$

Propozicija 1.1.22. *Metrika na unitarnom prostoru ima sljedeća svojstva:*

1. $d(x, y) \geq 0, \forall x, y \in V;$
2. $d(x, y) = 0 \Leftrightarrow x = y, \forall x, y \in V;$
3. $d(x, y) = d(y, x), \forall x, y \in V;$
4. $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in V.$

Definicija 1.1.23. *Neka je $X \neq \emptyset$. Svaka funkcija $d : X \times X \to \mathbb{R}$ sa svojstvima iz 1.1.22 naziva se metrika ili udaljenost. Tada (X, d) zovemo metrički prostor.*

1.2 Optimizacija

Problem određivanja ekstrema neke funkcije, uz zadane uvjete, naziva se problem matematičkog programiranja. Funkciju, kojoj je potrebno odrediti minimum ili maksimum, nazivamo funkcija cilja. Ukoliko je funkcija cilja linearna, te ako su uvjeti izraženi u obliku linearnih jednadžbi i/ili nejednadžbi, utoliko govorimo o problemu linearnog programiranja. Slično, kada je funkcija cilja kvadratna, govorimo o problemu kvadratnog programiranja.

Definicija 1.2.1. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup. Kažemo da funkcija $f : \Omega \to \mathbb{R}$ ima lokalni minimum u točki $P_0 \in \Omega$ ako postoji okolina $K(P_0, r) \subseteq \Omega$ takva da*

$$(\forall P \in K(P_0, r) \backslash P_0) \quad (f(P) \geq f(P_0)),$$

odnosno funkcija f u $P_0 \in \Omega$ ima lokalni maksimum ako vrijedi

$$(\forall P \in K(P_0, r) \backslash P_0) \quad (f(P) \leq f(P_0)).$$

Vrijednost $f(P_0)$ zovemo minimumom, odnosno maksimumom funkcije f na skupu Ω. Ako vrijede stroge nejednakosti, govorimo o strogom lokalnom minimumu, odnosno strogom lokalnom maksimumu. Ako nejednakosti vrijede za svaku početnu točku $P \in \Omega$, tada funkcija f u točki P_0 ima globalni minimum, odnosno globalni maksimum.

Definicija 1.2.2. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup i neka je $f : \Omega \to \mathbb{R}^n$ diferencijabilna funkcija. Za točku $P_0 \in \Omega$ kažemo da je stacionarna točka funkcije f ako vrijedi:*

$$\partial_i f(P_0) = 0, \quad i = 1, 2, ..., n.$$

Teorem 1.2.3. *(Nužni uvjet za postojanje lokalnog ekstrema) Ako je $P_0 \in \Omega \subseteq \mathbb{R}^n$ točka lokalnog ekstrema diferencijabilne funkcije $f : \Omega \to \mathbb{R}^n$, onda je P_0 stacionarna točka funkcije f, tj. vrijedi:*

$$\partial_i f(P_0) = 0, \quad i = 1, 2, ..., n.$$

Neka su zadane funkcije $f, g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, ..., m$. Promatramo sljedeći optimizacijski problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$g_i(x) \leq 0, \quad i = 1, 2, ..., m.$$

Skup $U = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \ i = 1, 2, ..., m\}$ zovemo *dopustivo područje*, a svaki $x \in U$ zovemo *dopustivo rješenje*. Dopustivo rješenje x^* za koje vrijedi $f(x^*) \leq f(x)$ zovemo *optimalno dopustivo rješenje*.

Gornjem problemu možemo pridružiti funkciju $L : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$ zadanu formulom

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x).$$

Funkciju L zovemo Lagrangeova funkcija koja je pridružena problemu.

Teorem 1.2.4. *Problem*

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$g_i(x) \leq 0, \quad i = 1, 2, ..., m$$

ekvivalentan je problemu

$$\min_{x \in \mathbb{R}^n} \max_{\alpha \in \mathbb{R}^m_+} L(x, \alpha).$$

Dokaz. Označimo s $g(x) := (g_1(x), ..., g_m(x))$. Uočimo da za fiksni $x \in \mathbb{R}^n$ vrijedi:

$$\max_{\alpha \in \mathbb{R}^m_+} L(x, \alpha) = \begin{cases} f(x), g(x) \leq 0 \\ \infty, inače \end{cases}.$$

Naime, za $g(x) \leq 0$ maksimum funkcije L po varijabli $\alpha \geq 0$ se postiže za $\alpha = 0$. S druge strane, ako je $g_i(x) > 0$ za neki $i \in \{1, ..., m\}$ povećanjem vrijednosti komponenata vektora $\alpha \in \mathbb{R}^m_+$ funkciju $L(x, \alpha)$ možemo proizvoljno povećati. Minimizacijom po $x \in \mathbb{R}^n$ vidimo da se minimum od $\max L(x, \alpha)$ postiže za $g(x) \leq 0$ i da je on jednak minimumu funkcije f na dopustivom skupu $U = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$, te su prema tome navedeni problemi uistinu ekvivalentni. □

Problem iz teorema zovemo *primarni problem*, budući da je rješenje primarnog problema ujedno rješenje originalnog optimizacijskog problema, njega također zovemo primarni problem. Možemo promatrati sljedeći optimizacijski problem

$$\max_{\alpha \in \mathbb{R}^m_+} \min_{x \in \mathbb{R}^n} L(x, \alpha)$$

kojeg zovemo dualni problem. Pretpostavimo da je zadan problem linearnog programiranja

$$\begin{cases} f(x) = c^T x \to min_x \\ Ax \geq b \end{cases}$$

pri čemu su $a \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$. Lagrangeovu funkciju $L : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$ definiramo formulom

$$L(x, \alpha) = c^T x + \alpha^T (b - Ax).$$

Tada su odgovarajući primarni i dualni problem

$$\min_{x\in\mathbb{R}^n} \max_{\alpha\in\mathbb{R}^m_+} L(x,\alpha),$$

$$\max_{\alpha\in\mathbb{R}^m_+} \min_{x\in\mathbb{R}^n} L(x,\alpha).$$

Poglavlje 2

Klasifikacija teksta

U posljednje vrijeme, zbog dostupnosti velike količine podataka i informacija, pojavilo se pitanje kako tu neprestano rastuću količinu dokumenata pretražiti i na osnovu nje dobiti što više upotrebljive informacije. Odabir bitne i upotrebljive informacije koja se isporučuje na korisnikov zahtjev može se poboljšati njezinom kategorizacijom. Budući da se radi o velikom rastu količine dokumenata, dolazi do razvoja metoda za automatsku klasifikaciju dokumenata prema sadržaju. Jedna od ključnih tehnika za organizaciju i upravljanje tekstualnim podacima jeste klasifikacija teksta. Smatra se jednom od najrazvijenijih disciplina na području umjetne inteligencije.

Klasifikacija teksta, kao vrsta nadziranog učenja, primjenjuje se u raznim domenama poput internet pretraživanja, filtriranja elektronske pošte, raspoznavanje rukopisa, govora ili klasifikacija članaka u određene kategorije. Cilj klasifikacije je automatiziranim postupkom pojedinom tekstu ili dokumentu odrediti klasu, pri čemu je klasa unaprijed definiran skup objekata s istim karakteristikama.

Za klasifikaciju teksta razvijene su brojne metode među koje ubrajamo i metodu osnovanu na semantičkom indeksiranju, tj. na postupku u kojem dokument opisujemo nizom reprezentativnih ključnih riječi čije značenje opisuje osnovnu tematiku promatranog dokumenta. Te ključne riječi nazivamo indeksnim pojmovima, koriste se za indeksiranje i sažimanje sadržaja dokumenta. Naime, ako uspoređujemo tekstove medicinske i tehničke tematike možemo primijetiti da postoje riječi koje su specifične za svaku od njih. Odnosno, u tekstovima medicinske tematike više će se pojavljivati riječi poput *"operacija"*, *"bolest"*, dok će tekstovi tehničke tematike više sadržavati riječi poput *"sila"*, *"električni"*. Indeksni pojmovi su većinom imenice jer one same po sebi imaju značenje, dok su pridjevi i prilozi u manjoj upotrebi jer većinom služe samo kao dopuna imenicama. U ovom pristupu postavljamo pitanje je li svaki indeksni pojam jednako koristan za klasifikaciju nekog dokumenta. Primjerice, ako želimo klasificirati 10 000 dokumenata i uočimo da se jedna riječ pojavljuje u svim dokumentima, zaključujemo da je ona manje značajna jer ne

pruža informaciju koja je korisna za klasifikaciju. Nadalje, riječ koja se pojavljuje u samo 10 dokumenata može znatno pomoći u klasifikaciji. Iz ovoga zaključujemo da važnost indeksnih pojmova nije jednaka. Ova se pojava u algoritmima za klasifikaciju prikazuje pomoću težinskih faktora. Sulton i Buckley su u [2] pokazali da sustav indeksiranja teksta, temeljen na težinskim faktorima, daje bolje rezultate prilikom klasifikacije dokumenata od onih dobivenih drugim, složenijim tekstualnim prikazima.

Da bi povećali uspješnost klasifikacije, dokumente je potrebno svesti na jednake forme te nakon toga formirati bazu reprezentativnih ključnih riječi koja će biti potrebna za klasifikaciju. Primjere načina pomoću kojih dobivamo bazu ključnih riječi opisujemo u sljedećim poglavljima.

2.1 Model *"Bag of words"*

Razvoj obrade i klasifikacije teksta tijekom godina je imao različite pristupe. U postupku obrade, dokumente je potrebno svesti na jednake forme. Neki postupci koji se primjenjuju u pripremi teksta su: uklanjanje brojeva, interpunkcijskih znakova, uklanjanje riječi koje ne pridaju značenju teksta (npr. veznici, prilozi, prijedlozi, itd). Nakon ovakve obrade teksta, zbog lakše klasifikacije, dokumente je potrebno prikazati kao niz ključnih riječi. Jedan od načina tvorbe ključnih riječi je korjenovanje.

Korjenovanje je postupak u kojemu se riječi skraćuju na korijen tako da se obriše završetak riječi, u većini slučajeva je to sufiks. Najčešći sufiksi u engleskom jeziku su *"-ed, -ions, -ion, -ing, -s"*. Cilj je skratiti riječ na korijen bez obzira na to je li taj korijen valjana riječ ili nije. Koristeći se algoritmom *Porter Stemmer*, opisan u izvoru [5], riječi *"program, programs, programmer, programming, programmers"* će se svesti na korijen *"program"*. Ovim postupkom dokument prikazujemo kao niz ključnih riječi koje ukazuju na njegov sadržaj.

U svrhu daljnje analize dokumenata, ideja klasifikacije je prikazati ih u obliku vektora značajki. Budući da smo korjenovanjem dokumente prikazali kao niz ključnih riječi, potrebno je formirati bazu ključnih riječi koja će predstavljati bazu vektorskog prostora, odnosno formiramo rječnik. Rječnik dobivamo tako što prolazimo kroz dokumente i izdvojimo svaku od ključnih riječi (bez ponavljanja), pri čemu poredak pojavljivanja riječi nije bitan. Ovako dobiven rječnik nazivamo *"bag of words"*.

Nakon postupka korjenovanja možemo formirati matricu koja opisuje dokumente pomoću frekvencija pojavljivanja pojmova iz rječnika. Tako dobivenu matricu nazivamo *document-term* matrica. Pri čemu i-ti red označava i-ti dokument, dok se u j-tom stupcu nalazi j-ti pojam iz baze riječi. Stoga se na (i, j)-tom mjestu matrice nalazi frekvencija j-tog pojma u i-tom dokumentu. Na ovako dobivenu matricu primjenjujemo neki od klasifikatora i analiziramo uspješnost klasifikacije.

Budući da informacije o frekvenciji pojavljivanja riječi iz baze u dokumentu ne mogu uvijek osigurati dobre rezultate, obično se uvode novi težinski faktori radi poboljšanja klasifikacije, primjerice:

1. *Inverzna frekvencija* - koristimo je u slučaju da se javljaju pojmovi s velikom frekvencijom u svim dokumentima neke kolekcije jer tada dobivamo da su svi dokumenti bitni i relevantni za taj pojam, što znatno pogoršava klasifikaciju. Uvođenjem inverzne frekvencije favoriziramo pojmove koji se nalaze u samo nekoliko dokumenata u kolekciji. Definiramo je kao logaritam omjera ukupnog broja dokumenata u kolekciji N i broja dokumenata u kojima se pojavljuje određeni pojam n:

$$\log \frac{N}{n}$$

2. *Faktor normalizacije*- koristimo u slučaju da se u kolekciji javljaju dokumenti različitih duljina. Dulji dokumenti imaju više riječi pa su tako i frekvencije veće. Kako je u cilju smanjiti ovisnost klasifikacije o duljini dokumenata, uvodi se faktor normalizacije kojeg definiramo kao omjer frekvencije pojma u nekom dokumentu i norme svih frekvencija pojmova tog dokumenta:

$$\frac{x_{i,j}}{||x_i||}, \textit{i=1, ..., broj dokumenata, j=1, ..., broj riječi u bazi}$$

2.2 Model *"Bag of n-grams"*

S obzirom na prije opisani pristup klasifikaciji dokumenata, postavljamo pitanje možemo li dovoljno dobro klasificirati dokumente ako zanemarimo postojanje riječi u dokumentu koje definiramo kao ograničen niz slova. Ukoliko je to moguće, utoliko nas zanima kolika su odstupanja u odnosu na model *"Bag of words"*. Na temelju postavljenog pitanja razvijamo model *"Bag of n-grams"*.

U modelu *"Bag of n-grams"*, nakon postupka svođenja tekstova na jednake forme, koji za razliku od poglavlja 2.1 uključuje samo uklanjanje separatora, brojeva te interpunkcijskih znakova, dobivamo dokumente u obliku niza slova. Tako dobivene dokumente dijelimo na *n-grame*, odnosno na odsječke veličine n neke riječi duljine k, pri čemu je $n \leq k$. Iz ovog proizlazi da nemamo rječnik, odnosno skup riječi koji čine dokument, nego ovim postupkom dobivamo rječnik čiji su elementi *n-grami*.

Neka je dokument, nakon svođenja na jednake forme, oblika:

LALAFALA...

Dakle, dokument je prikazan kao niz slova. Promotrimo 3-grame koji se pojavljuju u ovom dokumentu:

$$\begin{gathered} LALAFALA... \rightarrow LAL \\ LALAFALA... \rightarrow ALA \\ LALAFALA... \rightarrow LAF \\ \vdots \\ LALAFALA... \rightarrow ALA \end{gathered}$$

Slično, kao u modelu *"Bag od words"* kreiramo *"bag of n-grams"* , tj. formiramo listu *n-grama* koji su se pojavili u dokumentu (bez ponavljanja). Na takav način dobivamo bazu *n-grama* koji predstavljaju bazu vektorskog prostora. Stoga svaki dokument možemo prikazati kao vektor čiji su elementi frekvencija pojavljivanja određenog *n-grama* u pojedinom dokumentu. Dokument iz primjera prikazujemo kao vektor:

$$\begin{matrix} LAL \\ ALA \\ LAF \\ \vdots \end{matrix} \begin{bmatrix} 1 \\ 2 \\ 1 \\ \vdots \end{bmatrix} = v$$

Na ovakav način definiramo preslikavanje $\psi : \{dokument\} \rightarrow \mathbb{R}^m$, pri čemu je m dimenzija baze. Nakon provođenja postupka za sve dokumente, možemo formirati *document-term* matricu koja opisuje dokumente pomoću frekvencije pojavljivanja *n-grama* iz baze. Pri čemu *i*-ti red označava dokument, dok se u *j*-tom stupcu nalazi *j*-ti *n-gram* iz baze *n-grama*. Stoga se na *(i, j)*-tom mjestu matrice nalazi frekvencija *j*-tog *n-grama* u *i*-tom dokumentu. Kao i u prethodnom modelu, na ovako prikazane dokumente u obliku vektora značajki, možemo primijeniti neki od klasifikatora.

Uočimo kako u ovom modelu na klasifikaciju dokumenta može utjecati i duljina promatranog *n-grama*. Stoga bi bilo dobro promatrati uspješnost klasifikacije s obzirom na različite duljine *n-grama*.

Poglavlje 3

Stroj potpornih vektora

Stroj potpornih vektora *(SVM)* je skup metoda nadziranog učenja koje se koriste za probleme klasifikacije. To je algoritam koji, na temelju skupa podataka za treniranje, generira model koji određuje klasu testnog podatka. U daljnjem tekstu matematička podloga stroja potpornih vektora je preuzeta iz izvora [3].

3.1 Linearna klasifikacija

Binarna klasifikacija najčešće se izvodi pomoću realne funckije realne varijable $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ na sljedeći način: vektor $x = (x_1, ..., x_n)^T$ pripada pozitivnoj klasi ako je $f(x) \geq 0$, a negativnoj klasi ako je $f(x) < 0$. Promatramo linearnu funkciju $f(x)$ za $x \in X$, možemo je zapisati kao:

$$f(x) = \langle w, x \rangle + b = \sum_{i=1}^{n} w_i x_i + b$$

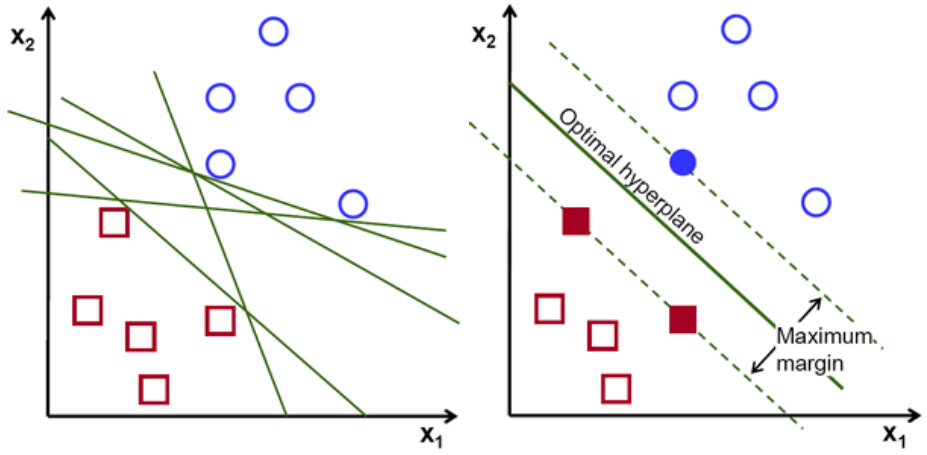
pri čemu su $\langle w, x \rangle \in \mathbb{R}^n \times \mathbb{R}$ parametri koji određuju funkciju. Slijedi da je geometrijska interpretacija ovakve funkcije takva da je prostor X podijeljen na dva poluprostora hiperravninom koja je određena jednadžbom $\langle w, x \rangle + b = 0$, pri čemu je w normala hiperravnine, a b predstavlja parametar za koji paralelno pomičemo tu hiperravninu. Parametar w nazivamo *vektor težine*, a *b pomak*.

Definicija 3.1.1. *Neka je $X \subseteq \mathbb{R}^n$ prostor ulaznih podataka te Y prostor rezultata. Skup učenja (eng. training set) je kolekcija primjera za učenje koju označavamo kao*

$$S = ((x_1, y_1), ..., (x_k, y_k)) \subseteq (X \times Y)^k$$

pri čemu je k broj primjera za učenje. Vektor x_i zovemo vektor značajki, a y_i njemu pridružena oznaka.

Za binarnu klasifikaciju vrijedi da je $Y = \{-1, 1\}$, dok za m-klasnu klasifikaciju vrijedi $Y = \{1, 2, ..., m\}$. Kažemo da je S *trivijalan* skup učenja ako svi primjeri imaju istu pridruženu oznaku. Reći ćemo da su primjeri *linearno odvojivi* ako postoji hiperravnina koja pravilno klasificira primjere za učenje. U suprotnom kažemo da primjeri nisu odvojivi. Takva hiperravnina nije jedinstvena. Cilj algoritma stroja potpornih vektora je klasificirati podatke, one koji su linearno odvojivi, na temelju optimalne hiperravnine.



Slika 3.1: Primjer mogućih hiperravnina i optimalne hiperravnine, izvor [7]

Kako bi pronašli optimalnu hiperravninu, uvodimo pojam *margine*. Definira se kao udaljenost pojmova za učenje određene klase do hiperravnine koja ih dijeli.

Definicija 3.1.2. *Funkcijska margina hiperravnine* (w, b) *s obzirom na primjer za učenje* (x_i, y_i) *dana je sa*

$$y_i = y_i(\langle w, x_i \rangle + b).$$

Definicija 3.1.3. *Funkcijska margina hiperravnine* (w, b) *s obzirom na skup učenje* S *dana je sa* $y = \min_i y_i$.

Cilj je imati što veću funkcijsku marginu. Ako skaliramo hiperravninu (w, b) za željeni skalar $\lambda \in \mathbb{R}^+$ dobivamo hiperravninu $(\lambda w, \lambda b)$. Za mjerenje euklidske udaljenosti primjera za učenje od linearnog klasifikatora u skupu za učenje koristimo *geometrijsku marginu*. Definiramo je kao funkcijsku marginu normirane hiperravnine $(\frac{1}{\|w\|}w, \frac{1}{\|b\|}b)$. Stoga je *margina skupa za učenje* S maksimalna geometrijska margina svih mogućih razdvajajućih hiperravnina. Optimizacija geometrijske margine se provodi tako što postavimo funkcionalnu

marginu na vrijednost $y = 1$ i minimiziramo normu vektora težine w. Matematički zapisujemo

$$\begin{cases} \langle w, w \rangle \to \min_{w,b} \\ y_i(\langle w, x_i \rangle + b) \geq 1, \\ i = 1, ..., k \end{cases}$$

Ovaj optimizacijski problem nazivamo primarna forma.
Postavimo li vektor težine w tako da funkcijska margina ima vrijednost $y = 1$ za pozitivan primjer x^+ i negativan primjer x^-, takva funkcijska margina implicira

$$\langle w, x^+ \rangle + b = +1$$

$$\langle w, x^- \rangle + b = -1$$

dok za geometrijsku marginu moramo normirati vektor w. Tada je geometrijska margina y dana s

$$y = \frac{1}{2}\left(\left\langle \frac{w}{||w||_2}, x^+ \right\rangle - \left\langle \frac{w}{||w||_2}, x^- \right\rangle\right) = \frac{1}{2||w||_2}\left(\langle w, x^+ \rangle - \langle w, x^- \rangle\right) = \frac{1}{||w||_2}.$$

Time smo pokazali sljedeću propoziciju:

Propozicija 3.1.4. *Neka je dan linearno odvojiv skup podataka za trening*

$$S = ((x_1, y_1), ..., (x_k, y_k)).$$

Hiperravnina (w, b) *koja rješava optimizacijski problem*

$$\begin{cases} \langle w, w \rangle \to \min_{w,b} \\ y_i(\langle w, x_i \rangle + b) \geq 1, \\ i = 1, ..., k \end{cases}$$

ima maksimalnu marginu hiperravnine kao geometrijsku marginu $y = \frac{1}{||w||_2}$.

Podaci za trening x_i koji su najbliže hiperravnini su upravo oni za koje je funkcijska margina jednaka $y = 1$, stoga ih zovemo *potporni vektori*.
Optimizacijski problem se rješava uz pomoć Lagrangeovih multiplikatora. Stoga prethodni problem prikazujemo u obliku:

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

pri čemu su $\alpha \geq 0$ Lagrangeovi multiplikatori. Prvi dio izraza predstavlja funkciju koju želimo minimizirati, a drugi dio predstavlja uvjet koji mora biti zadovoljen. Optimizacijski

problem je sada sveden na traženje minimuma Lagrangeove funkcije. Nužan uvjet za ekstrem funkcije, u ovom slučaju minimuma, dobivamo deriviranjem Lagrangeove funkcije po parametrima w i b te izjednačavanjem parcijalnih derivacija s nulom:

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{k} y_i \alpha_i x_i = 0 \Rightarrow w = \sum_{i=1}^{k} y_i \alpha_i x_i$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{k} y_i \alpha_i = 0$$

Slijedi

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

Time smo pokazali sljedeću propoziciju:

Propozicija 3.1.5. *Neka je dan linearno odvojiv skup podataka za trening*

$$S = ((x_1, y_1), ..., (x_k, y_k)),$$

i pretpostavimo da parametri α^* rješavaju sljedeći optimizacijski problem:

$$\begin{cases} \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \to \max, \\ \sum_{i=1}^{k} y_i \alpha_i = 0, \alpha_i \geq 0, i = 1, ..., k. \end{cases}$$

Tada vektor težine $w^* = \sum_{i=1}^{k} y_i \alpha_i^* x_i$ određuje maksimalnu marginu hiperravnine čija je geometrijska margina $\gamma = \frac{1}{\|w\|_2}$.

Poglavlje 4

Obrada i klasifikacija teksta pomoću *n-grama*

Budući da se posljednjih godina povećala potreba za organizacijom informacija, posljedično se javio veliki interes za klasifikacijom tekstualnih dokumenata, tj. svaki tekstualni dokument se želi pridružiti nekoj postojećoj klasi.

U ovom radu ćemo promatrati i analizirati dvije kolekcije članaka te ispitati točnost klasifikacije koja se temelji na modelu *"Bag of n-grams"*, pri čemu je *n* varijabilan. Promatrat ćemo slučajeve: 1 do 3-gram, 1 do 4-gram, 4-gram, 1 do 5-gram i 5-gram. Također ćemo usporediti rezultate dobivene ovim modelom i rezultate klasifikacije dobivene modelom *"Bag of words"* koje čitamo iz izvora [5].

4.1 Set podataka i reprezentacija teksta

Kao što smo naveli, promatramo dvije kolekcije članaka na engleskom jeziku: CRAN i MED. CRAN kolekcija se sastoji od 1398 članaka čije su teme vezane za aeronautiku te su prikupljeni na sveučilištu u Cranfieldu, dok je MED kolekcija od 1033 članaka medicinske tematike izvučenih iz Medline časopisa. Obje kolekcije članaka su preuzete sa servera odsjeka računalnih znanosti sveučilišta u Glasgowu [7]. Primjer takvog jednog članka je prikazan na slici 4.1. U daljnjem tekstu kolekcije ćemo nazivati *klasama*, a članke *dokumentima*.

Da bi klasificirali dokumente, potrebno ih je svesti na jednake forme. Kako programski jezici razlikuju velika i mala slova, potrebno je proći kroz sve dokumente i velika slova zamijeniti malim slovima. Budući da interpunkcijski znakovi, brojevi i oznake za novi red ne daju informaciju o pripadnosti dokumenta nekoj od klasa, njih uklanjamo. U izvoru [5] nakon ovakve obrade dokumenata dodatno se uklanjaju riječi koje ne pridonose značenju teksta kao što su prilozi, prijedlozi, veznici i sl. te se tako formirani dokument prikazuje

kao niz ključnih riječi dobivenih postupkom korjenovanja (eng. *stemming*), slika 4.2. Dok ćemo u ovom diplomskom radu, nakon obrade, dokumente prikazati kao niz slova. Dakle, samo ćemo još ukloniti separatore, slika 4.3.

```
.I 1
.T
18 Editions of the Dewey decimal Classifications
.A
Comaromi, J.P.
.W
The present study is a history of the DEWEY Decimal Classification.  The first
edition of the DDC was published in 1876, the eighteenth edition in 1971, and
future editions will continue to appear as needed.  In spite of the DDC's long
and healthy life, however, its full story has never been told.  There have been
biographies of Dewey that briefly describe his system, but this is the first
attempt to provide a detailed history of the work that more than any other has
spurred the growth of librarianship in this country and abroad.
```

Slika 4.1: Primjer dokumenta prije obrade teksta

```
present studi histori dewey decim classif first edit ddc publish eighteenth edit
futur edit continu appear need spite ddc long healthi life howev full stori never
told biographi dewey briefli describ system first attempt provid detail histori
work spur growth librarianship countri abroad
```

Slika 4.2: Obrada dokumenta za model *"Bag of words"*

```
thepresentstudyisahistoryofthedeweydecimalclassificationthefirsteditionoftheddc-
waspublishedintheeighteenthedition inandfutureeditionswillcontinuetoappear-
asneededinspiteoftheddcslongandhealthylifehoweveritsfullstoryhasneverbeen-
toldtherehavebeenbiographiesofdeweythatbrieflydescribehissystembutthisisthe-
firstattempttoprovideadetailedhistoryoftheworkthatmorethananyotherhasspurredthe-
growthoflibrarianshipinthiscountryandabroad
```

Slika 4.3: Obrada dokumenta za model *"Bag of n-grams"*

Na ovaj način smo formirali svaki od 2431 dokumenata kao niz slova, njih 1398 iz CRAN klase, a 1033 iz MED klase. Ideja je stvoriti vektor značajki za svaki pojedini dokument. Kao što smo naveli u poglavlju 2.2, zbog nedostatka separatora nemamo rječnik,

odnosno bazu ključnih riječi koje se pojavljuju u dokumentima (bez ponavljanja), nego stvaramo vlastiti n-rječnik koji se sastoji od *n-grama* (bez ponavljanja). Promatranjem različitih varijanti *n-grama*, dobivamo n-rječnike:

(1) Za 1 do 3-grame dobivamo n-rječnik koji se sastoji od 9121 n-grama, pri čemu je n=1, 2, 3.

(2) Za 1 do 4-grame dobivamo n-rječnik koji se sastoji od 61005 n-grama, pri čemu je n=1, 2, 3, 4.

(3) Za 4-grame dobivamo 4-rječnik koji se sastoji od 51884 4-grama.

(4) Za 1 do 5-grame dobivamo n-rječnik koji se sastoji od 229662 n-grama, pri čemu je n=1, 2, 3, 4, 5.

(5) Za 5-grame dobivamo 5-rječnik koji se sastoji od 168658 5-grama.

4.2 Document-term matrica

Nakon generiranja n-rječnika, svaki pojedini dokument smo prikazali kao vektor značajki pomoću *n-grama* iz rječnika. Time smo formirali *document-term matricu* koja opisuje dokumente pomoću frekvencija *n-grama* i u kojoj svaki redak prestavlja jedan dokument, a svaki stupac predstavlja jedan *n-gram* iz baze. Dakle, dobivena je matrica dimenzije $2431 \times k$, gdje je k dimenzija baze, pri čemu i-ti red označava dokument, dok j-ti stupac predstavlja *n-gram* iz baze. Stoga se na *(i, j)*- tom mjestu nalazi frekvencija pojavljivanja j-tog pojma u i-tom dokumentu. Slijedi, i- ti redak $x_i = (x_{i,1}, x_{i,2}, ...x_{i,k})$ predstavlja vektor značajki za i-ti dokument, i=1, 2, ..., 2431, k=dimenzija baze. Kako smo promatrali dokumente iz dvije klase, prvih 1398 redaka matrice su dokumenti iz CRAN klase, a preostali reci su dokumenti iz MED klase.

Budući da smo promatrali različite duljine *n-grama*, tako smo formirali *document-term* matrice različitih dimenzija:

(1) Za 1 do 3-grame smo dobili matricu dimenzija 2431×9121.

(2) Za 1 do 4-grame smo dobili matricu dimenzija 2431×61005.

(3) Za 4-grame smo dobili matricu dimenzija 2431×51884.

(4) Za 1 do 5-grame smo dobili matricu dimenzija 2431×229662.

(5) Za 5-grame smo dobili matricu dimenzija 2431×168658.

Također smo formirali vektor oznaka, duljine 2431, pri čemu na *i*- tom mjestu stoji oznaka kojoj klasi *i*-ti dokument pripada: 0 ako je dokument iz klase CRAN, a 1 ako dokument pripada klasi MED.

4.3 Primjena SVM i rezultati

Kako smo svaki dokument prikazali kao vektor značajki pomoću *n-grama* iz rječnika, potrebno ih je podijeliti na trening i test skup. Trening skup nam služi da bi izgradili i analizirali rješenje, a test skup za testiranje rješenja i procjenu točnosti rezultata. Na skupu za učenje ćemo generirati model pomoću algoritma stroja potpornih vektora koji određuje kojoj klasi pripada testni dokument. Imajući u vidu da su dokumenti u trening skupu grupirani u klase, želimo provjeriti koliko dobro algoritam stroja potpornih vektora može klasificirati dokumente iz test skupa. Kako znamo kojoj klasi pripada testni dokument, lako možemo pratiti koliko je uspješna klasifikacija.

Za procjenu uspješnosti klasifikacije, potrebno je standardizirati određene pokazatelje. Da bi definirali mjere uspješnosti potrebno je definirati pojmove: točni pozitivni (*eng. true positives TP*), lažni pozitivni (*eng. false positives FP*), točni negativni (*eng. true negatives TN*) i lažni negativni (*eng. false negatives FN*).

Točni pozitivni (*TP*) definirani su kao broj ispravno klasificiranih dokumenta u točnu klasu kojoj pripadaju.

Lažni pozitivni (*FP*) definirani su kao broj dokumenata koji su svrstani u klasu u kojoj zapravo ne pripadaju.

Točni negativni (*TN*) definirani su kao broj dokumenata kod kojih je točno ustanovljeno da ne pripadaju određenoj klasi.

Lažni negativni (*FN*) definirani su kao broj dokumenata koji bi u stvarnosti pripadali određenoj klasi, ali ih model nije klasificirao u tu istu klasu.

		Actual Class	
		Positive (**P**)	Negative (**N**)
Predicted Class	Positive (**P**)	True Positive (**TP**)	False Positive (**FP**)
	Negative (**N**)	False Negative (**FN**)	True Negative (**TN**)

Slika 4.4: Matrica zabune

Slijedi, mjere uspješnosti koje promatramo su: preciznost (eng. *precision*) i odziv (eng. *recall*). Preciznost računa omjer između dokumenata koji su točno klasificirani u neku

klasu i ukupnog broja dokumenata klasificiranih u tu klasu, a odziv nam daje informaciju koliko je dokumenta točno klasificirano u određenu klasu u odnosu na ukupan broj dokumenata koji su trebali biti klasificirani u tu klasu. U cilju nam je imati visok odziv kako bi broj točno klasificiranih dokumenata bio što veći i visoku preciznost kako bi broj krivo klasificiranih dokumenata bio što manji. Iz ove dvije mjere proizlazi mjera točnost. To je mjera uspješnosti koju definiramo kao omjer točno klasificiranih dokumenata i ukupnog broja dokumenata.

$$\begin{aligned} preciznost &= \frac{TP}{TP+FP} \\ odziv &= \frac{TP}{TP+FN} \\ točnost &= \frac{TP+TN}{TP+TN+FP+FN} \end{aligned}$$

Kako promatramo *n-grame* različitih duljina, testiranje smo proveli za svaku varijantu. U svim slučajevima smo podijelili podatke na trening i test skup u omjeru 50 : 50. Podjelu podataka smo izveli pet puta za svaku varijantu *n-grama*. Testiranje provodimo tako što dokumente iz testnog seta uvrštavamo u algoritam stroja potpornih vektora. Budući da znamo klasu kojoj određeni testni dokumenti pripada, gledamo koliko je dokumenata algoritam klasificirao u pravu klasu. Dobiveni rezultati su:

	1 do 3-gram	1 do 4-gram	4-gram	1 do 5-gram	5-gram
1)	0.9423	0.9497	0.8707	0.8707	0.5810
2)	0.9547	0.9292	0.8905	0.8600	0.5851
3)	0.9522	0.9679	0.9202	0.8411	0.5679
4)	0.9456	0.9572	0.9037	0.8691	0.5868
5)	0.9382	0.9547	0.9243	0.8847	0.5654
Prosjek	**0.9466**	**0.9517**	**0.9091**	**0.8651**	**0.5772**

Tablica 4.1: Točnost klasifikacije SVM-a

U usporedbi varijanti *n-grama* uočavamo da najveću točnost dobivamo za 1 do 4-grame, ona iznosi u prosjeku 95.17%. Točnost za 1 do 5-grame je za 8.66% manja, stoga zaključujemo da se dodavanjem duljih *n-grama* smanjuje točnost klasifikacije. Također, točnost klasifikacije temeljene na 4-gramima iznosi 90.91% što znači da dodavanjem kratkih *n-grama* poboljšavamo klasifikaciju.

Ukoliko našu najbolju točnost usporedimo s točnošću klasifikacije tekstulanih dokumenata koja se temelji na korjenovanju riječi, gdje je točnost 98.60%, izvor [5], utoliko

zaključujemo da klasifikacija u kojoj promatramo 1 do 4-grame gotovo jednako dobro klasificira dokumente. Dakle, ovim testom smo pokazali da znanje o riječima koje se nalaze u tekstualnim dokumentima nema veliku značajnost jer promatranjem 1 do 4-grama dobivamo odstupanje od 3.43% što je i dalje dovoljno dobra klasifikacija.

Uočili smo kako frekvencije pojavljivanja 1 do 4-grama u dokumentima osiguravaju dobre informacije, u cilju nam je provjeriti možemo li točnost klasifikacije poboljšati tako što modificiramo *document-term* matricu. Kao što smo naveli u poglavlju 2, modifikaciju ćemo provesti tako što ćemo svaki element modificirane matrice dobiti pomoću formule:

$$\log \frac{opažena\ frekvencija + 1}{ukupna\ frekvencija + 1}$$

pri čemu je *opažena frekvencija* broj pojavljivanja *j*-tog *n-grama*, u *i*-tom dokumentu, a *ukupna frekvencija* je ukupan broj pojavljivanja *j*-tog *n-grama* u svim dokumentima, za n=1, 2, 3, 4.

Primjenom algoritma stroja potpornih vektora na ovako modificiranu matricu dobivamo rezultate:

	1 do 3-gram	1 do 4-gram	4-gram	1 do 5-gram	5-gram
Točnost	0.5728	0.5811	0.5662	0.5876	0.5860

Tablica 4.2: Točnost klasifikacije SVM-a primijenjen na modificiranu matricu

Zaključujemo da je točnost klasifikacije znatno lošija iz čega slijedi da postupak modificiranja *document-term* matrice ne osigurava poboljšanje klasifikacije u ovom slučaju.

Na kraju, možemo reći da klasifikacija za naše podatke, koja se temelji na modelu *"Bag od n-grams"* u kojem promatrmo 1 do 4-grame, gotovo jednako dobro klasificira dokumente kao klasifikacija u kojoj smo dokumente prikazali kao niz korijena riječi. Iz ovoga zaključujemo da postupak korjenovanja riječi nema značajnu ulogu u klasifikaciji dokumenta koje promatramo. Također, primjećujemo kako su rezultati klasifikacije u modelu *"Bag od n-grams"* bolji ako se uključe *n-grami* malih duljina, pri čemu je $n \leq 4$, od onih u kojima je $n = 5$.

Poglavlje 5

Python implementacija

Za implementaciju klasifikacije tekstualnih dokumenata koristili smo se Python-ovim *scikit-learn* paketima, kao što su *SciPy* i *NumPy*. Koristeći se Python-ovom funkcijom *CountVectorize*, kolekcije tekstualnih dokumenata smo prikazali kao matricu čiji su elementi frekvencije ključnih riječi iz baze. Ovim postupkom smo dobili matricu velikih dimenzija koja sadrži jako puno nula. Takva vrsta matrice se naziva rijetka matrica *(eng. sparse matrix)*. Budući da je matrica velikih dimenzija, slijedi da zauzima radnu memoriju. Jedan od načina kako smo oslobodili radnu memoriju i ubrzali proces je korištenje Python-ove funkcije *sparse*; matricu smo prikazali samo pomoću elemenata koji su različiti od 0. Na tako dobivenu matricu smo primijenili klasifikator stroja potpornih vektora, u Pythonu je to funkcija *svm*.

Bibliografija

[1] D. Bakić, *Linearna algebra*, Školska knjiga, Zagreb, 2008.

[2] C. Buckley, G. Salton, *Term-weighting approaches in automatic text retrieval*, In Information Processing and Management, Volume 24, Issue 5, 1988.

[3] F. Janjić, *Semantičko indeksiranje i klasifikacija dokumenata*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek, 2019.

[4] M. Pezić, *Tehnike učenja za klasifikaciju bioloških nizova*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek, 2020

[5] T. Rumec, *Varijante semantičkog indeksiranja i klasifikacija dokumenata*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek, 2020.

[6] R. Gandhi, *Support Vector Machine: Complete theory*, dostupno na `https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47`

[7] Glasgow IDOM, dostupno na `https://ir.dcs.gla.ac.uk/resources/test_collections`

Sažetak

Cilj rada je klasifikacija tekstualnih dokumenata pomoću algoritma stroja potpornih vektora i n-gramskog rječnika. Prvo smo definirali pojmove iz linearne algebre i optimizacije koji su potrebni za razumijevanje obrađenih tema u radu. Objašnjen je pojam klasifikacije teksta te pristupi: *"Bag of words"* i *"Bag of n-grams"*. Također je opisana matematička pozadina algoritma stroja potpornih vektora te njegova implementacija u programskom jeziku Python. Za analizu smo koristili dvije kolekcije tekstualnih dokumenata, pri čemu smo trening i test skup podijelili u omjeru 50 : 50. Nakon što smo napravili rječnik *n-grama*, proveli smo testiranje te smo usporedili rezultate klasifikacije za navedene pristupe.

Summary

The aim of this work is to classify text documents using the support vector machine algorithm and the n-gram dictionary. First, we present basic concepts from linear algebra and optimization theory required for the understanding of topics covered in the paper. Also, we explained the concept of text classification and approaches: *"Bag of words"* and *"Bag of n-grams"*. Next, the mathematical background for support vector machine algorithm is described with its implementation in Python programming language. We used two sets of text documents for analysis, dividing the training set and test set into 50 : 50 ratio. After creating the n-gram dictionary, we conducted the test and compared the classification results for the approaches listed above.

Životopis

Rođena sam 14.09.1994. godine u Novoj Gradiški. Osnovnu školu Matija Gubec Cernik upisujem 2001. godine koju završavam 2009. godine. Obrazovanje nastavljam upisom u Opću gimaziju Nova Gradiška. Godine 2013. završavam srednjoškolsko obrazovanje i upisujem Preddiplomski studij Matematike i informatike na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti u Mostaru, BIH, kojeg završavam 2018. godine. Iste godine upisujem Diplomski sveučilišni studij Matematičke statistike na Prirodoslovno-matematičkom fakultetu u Zagrebu.