**Univerzitet u Nišu**

**Elektronski fakultet**

**Katedra za računarstvo**

**Procesiranje prirodnih jezika**

**Izveštaj domaćeg zadatka**

**Student:**

**Stefan Stojanović 1355**

**Mentor:**

**Prof. dr Suzana Stojković**

**Tema domaćeg zadatka : Shakespeare classification**

**Korišćeni dataset-ovi**

Za izradu domaćeg zadatka korišćena su dva dataseta. Jedan dataset sardži korpus rečenica iz svih Šekspirovih dela.

Link do dataseta: <https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays>

Ovaj dataset sadrži 54 990 rečenica.

Za treniranje je korišćen podskup od izabranih 5000 rečenica.

Drugi dataset sadrži korpus rečenica iz svakodnevnog govora na modernom engleskom jeziku.

Link do dataseta: <https://github.com/jimbozhang/speechocean762>

U ovom datasetu se nalazi 2500 rečenica kao deo training seta I 2500 rečenica kao deo test seta. Za izradu domaćeg zadatka sam spojio ova dva seta rečenica u jedan.

**Izrada domaćeg zadatka**

**#1 Preprocesiranje dataseta**

Pre izrade samog zadatka izvršeno je prečišćavanje dataset-ova koji će se koristiti zbog puno nepotrebnih linija I fraza. Obzirom na to da su šekspirova dela drame, dataset sadrži linije poput SCENE I, Enter KING HENRY, Queen and others, Prologue, [neki tekst] itd. Ove fraze se veoma često javljaju u datasetu pa mogu dovesti do pogrešnih rezultata kod treniranja klasifikatora.

U datasetu modernih rečenica svaka rečenica je zauzimala jednu liniju txt fajla ali se nijedna od njih nije završavala tačkom, pa je bilo potrebno dodati tačke na kraju svake rečenice.

**#2 Preprocesiranje teksta**

Korišćenjem PlaintextCorpusReader-a učitavamo oba korpusa rečenica u promenljivama shakespeare\_text i modern\_text u raw formatu.

Pomoću sent\_tokenize delimo ulazne korpuse na rečenice i čuvamo ih u shakespare\_sentences i modern\_sentences.

Nakon podele tekstova na rečenice primenjujemo identičan skup radnji za oba skupa rečenica u dve odvojene for petlje.

sent = re.sub('[^A-Za-z]', ' ', sentence) #remove non-text

sent = sent.lower() #lower all letters

sent = word\_tokenize(sent) #tokenize sentence

Za svaku rečenicu izvršavamo sledeći skup radnji:

* izbacivanje svih znakova koji nisu slova;
* zamena velikih slova malim;
* tokenizacija rečenice na tokene tj. reči.

Za svaki token u jednoj rečenici izvršavamo sledeći skup radnji:

* propuštanje tokena kroz speller kako bi ispravio eventualne greške;
* proveru da li je token stopword;

Ukoliko token nije stopword u jednom slučaju token propuštamo kroz Porterov stemer, a u drugom slučaju vršimo lematizaciju I token propuštamo kroz WordNetLemmatizer. Sve procesirane tokene zatim spajamo pomoću blanko znakova između svaka dva tokena u procesiranu rečenicu I u zavisnosti od slučaja I primenjenih operacija smeštamo rečenicu u jednu od 4 liste: processed\_shakespeare\_sentences\_s, processed\_shakespeare\_sentences\_l, processed\_modern\_sentences\_s, processed\_modern\_sentences\_l. Na kraju spajamo ove 4 liste u 2 liste: processed\_sentences\_s I processed\_sentences\_l.

**#3 Ekstrakcija atributa**

Nakon preprocesiranja ulaznog teksta možemo ivršiti ekstrakciju atributa korišćenjem Bag of Words pristupa. Isprobaćemo oba pristupa tj. korišćenje TF mere (CountVectorizer) i korišćenje TF-IDF mere(TfidfVectorizer). Za maksimalan broj feature-a uzimamo 250.

**TF mera**

Kreiramo matrice:

X\_s - matrica za rečenice i stemer

X\_l - matrica za rečenice i lematizaciju

Kreiramo listu:

y - sadrži labelu 'shakespeare' onoliko koliko ima Šekspirovih rečenica i labelu 'modern' onoliko koliko ima modernih rečenica

**TF-IDF mera**

text\_tf\_s = tf\_s.fit\_transform(processed\_sentences\_s)

text\_tf\_l = tf\_l.fit\_transform(processed\_ sentences\_l)

Korišćenjem TfidfVectorizer dobili smo dve retko posednute matrice za stemer i lematizaciju.

**#3 Izbor klasifikatora i treniranje**

Koristićemo Naivni Bajesov klasifikator, DecisionTreeClassificator, MaxentClassificator. Da bismo trenirali podatke potrebno je da generišemo podatke u obliku (features, label) gde je features dictionary u kojem ključ predstavlja naziv atributa a value vrednost tog atributa.

Za korišćenje TF mere ove podatke generišemo u okviru X\_nltk\_s i X\_nltk\_l promenljivih.

Za korišćenje TF-IDF mere ove podatke generišemo u okviru data\_tf\_s i data\_tf\_l promenljivih

Podaci za klasifikaciju su trenutno rasporedjeni u redosledu Šekspirovi podaci - Moderni podaci i potrebno je promešati ove podatke kako bismo dobili dobre trening i test setove. Pomoću biblioteke random mešamo podatke i delimo ih na dva jednaka dela.

train\_set\_s, test\_set\_s = X\_nltk\_s[half\_of\_entries:], X\_nltk\_s[:half\_of\_entries] - trening podaci koriščenjem TF mere i stemera

train\_set\_l, test\_set\_l = X\_nltk\_l[half\_of\_entries:], X\_nltk\_l[:half\_of\_entries] -trening podaci korišćenjem TF mere i lematizacije

train\_set\_tf\_s, test\_set\_tf\_s = data\_tf\_s[half\_of\_entries:], data\_tf\_s[:half\_of\_entries] - trening podaci korišćenjem TF-IDF mere i stemera

train\_set\_tf\_l, test\_set\_tf\_l = data\_tf\_l[half\_of\_entries:], data\_tf\_l[:half\_of\_entries] - trening podaci korišćenjem TF-IDF mere i lematizacije

Treniramo 12 klasifikatora:

classifierNLTK\_s = NaiveBayesClassifier.train(train\_set\_s) - Bajesov klasifikator u kojem je korišćena TF mera i stemer

classifierNLTK\_l = NaiveBayesClassifier.train(train\_set\_l) - Bajesov klasifikator u kojem je korišćena TF mera i lematizacija

classifier\_s = nltk.classify.NaiveBayesClassifier.train(train\_set\_tf\_s) - Bajesov klasifikator u kojem je korišćena TF-IDF mera i stemer

classifier\_l = nltk.classify.NaiveBayesClassifier.train(train\_set\_tf\_l) - Bajesov klasifikator u kojem je korišćena TF-IDF mera i lematizacija.

classifierDT\_s = nltk.classify.DecisionTreeClassifier.train(train\_set\_s) - DecisionTree klasifikator u kojem je korišćena TF mera i stemer

classifierDT\_l = nltk.classify.DecisionTreeClassifier.train(train\_set\_l) DecisionTree klasifikator u kojem je korišćena TF mera i lematizacija.

classifierME\_s = nltk.classify.MaxentClassifier.train(train\_set\_s, trace=0) Maxent klasifikator u kojem je korišćena TF mera i stemer

classifierME\_l = nltk.classify.MaxentClassifier.train(train\_set\_l, trace=0) Maxent klasifikator u kojem je korišćena TF mera i lematizacija.

classifierDT\_s\_IDF = nltk.classify.DecisionTreeClassifier.train(train\_set\_tf\_s) DecisionTree klasifikator u kojem je korišćena TF-IDF mera i stemer

classifierDT\_l\_IDF = nltk.classify.DecisionTreeClassifier.train(train\_set\_tf\_l) DecisionTree klasifikator u kojem je korišćena TF-IDF mera i lematizacija.

classifierME\_s\_IDF = nltk.classify.MaxentClassifier.train(train\_set\_tf\_s, trace=0) Maxent klasifikator u kojem je korišćena TF-IDF mera i stemer

classifierME\_l\_IDF = nltk.classify.MaxentClassifier.train(train\_set\_tf\_l, trace=0) Maxent klasifikator u kojem je korišćena TF-IDF mera i lematizacija.

**Rezultati**

Korišćenjem funkcije nltk.classify.accuracy na odredjenim test skupovima koje smo generisali za svaki slučaj proveravamo tačnost treniranih klasifikatora. Rezultate ćemo prikazati u okviru tabele.

|  |  |
| --- | --- |
| Korpus rečenica  (klasifikator,  stem-lem, TF/TF-IDF | Jednak broj rečenica |
| (Bajes, stemer, TF) | 83.610% |
| (Bajes-lematizacija, TF) | 83.990% |
| (Bajes, stemer, TF-IDF) | 52.071% |
| (Bajes, lematizacija, TF-IDF) | 53.472% |
| (DecisionTree, stemer, TF) | 82.069% |
| (DecisionTree, lematizacija, TF) | 83.330% |
| (DecisionTree, stemer, TF-IDF) | 54.613% |
| (DecisionTree, lematizacija, TF-IDF) | 58.755% |
| (Maxent, stemer, TF) | 78.727% |
| (Maxent, lematizacija, TF) | 78.087% |
| (Maxent, stemer, TF-IDF) | 54.813% |
| (Maxent, lematizacija, TF-IDF) | 55.393% |

Na osnovu dobijenih rezultata možemo zaključiti da korišćenje lematizacije za dati problem daje neznatno bolje rezultate, kao i da korišćenje TF mere u odnosu na TF-IDF meru daje znatno bolje rezultate. Za dati problem najbolje rezultate daje Naivni Bajesov klasifikator. Izvršićemo proveru na nekim rečenicama.

**Test rečenice**

Za testiranje ćemo koristiti klasifikator koji ima najveći accuracy. To je Naivni Bajesov Klasifikator uz korišćenje lematizacije i TF mere.

Testiranje je izvršeno na sledećem skupu rečenica:

#shakespeare

test\_sentences1 = We will die all three: But I will prove that two kings are as good As him and me. '

#modern

test\_sentences2 = 'Hello, how are you and what are you doing today, and will you go out with friends tonight? '

#modern

test\_sentences3 = 'One girl going to see nine dolphins. '

#modern

test\_sentences4 = 'In that he went too far away. '

#modern

test\_sentences5 = 'AUTHORITIES CLAIMS THAT THE KILLINGS WERE THE crime OF ANGER. '

#modern

test\_sentences6 = 'I was very little when i lost my tooth. '

#shakespeare

test\_sentences7 = 'But, gracious sir, Here are your sons again, and I must lose Two of the sweet\'st companions in the world.'

Da bi se vršilo testiranje potrebno je preprocesirati ulazne rečenice I klasifikatoru proslediti dictionary za svaku rečenicu u obliku (features, values). Funkcija test\_sentences\_func preprocesira ulazni tekst, deli ga na rečenice, kreira dictionary-e I poziva klasifikator za svaku izdvojenu rečenicu tako što prosleđuje generisani dictionary za tu rečenicu. Rezultat je u nastavku.

Sentence number 1 is shakespeare

Sentence number 2 is modern

Sentence number 3 is modern

Sentence number 4 is modern

Sentence number 5 is modern

Sentence number 6 is modern

Sentence number 7 is shakespeare

Klasifikator je tačno odredio sve rečenice.

Pokušaćemo da kreiramo primere u kojima klasifikator neće ispravno odrediti tip rečenice.

#modern

test\_sentences8 = 'King George didn\'t fear anything and decided that he is going to see that spectacle. '

#modern

test\_sentences9 = 'George six didn\'t fear anything and he looked at something. '

#modern

test\_sentences10= 'Panmure stabbing: Bradford Kipa pleads guilty to murder of John Tofu Ioane.'

#modern

test\_sentences11= 'Panmure stabbing: Bradford Kipa pleads guilty to murder of John Tofu Ioane which died. '

Sentence number 8 is shakespeare

Sentence number 9 is modern

Sentence number 10 is modern

Sentence number 11 is shakespeare

8. rečenicu klasifikator je pogrešno protumačio zbog reči king i fear koje se najčešće pojavljuju u Šekspirovim rečenicama.

9. rečenicu klasifikator je tačno protumačio jer ima jednu reč manje koja pogoduje Šekspirovom stilu(king) a reč koja je odlika modernog teksta looked se nalazi u ovoj rečenici.

10. rečenica je moderan naslov članka i tačno je odredjena.

11. rečenica je modifikacija 10. rečenice sa dodatkom died na kraju i klasifikator ovu rečenicu tumači kao Sekspirovu. Razlog tome je sto se u Šekspirovim rečenicama veoma često spominje reč die pa ova reč ima veliku težinu pri odlučivanju.

Ukoliko izvršimo sledeću naredbu classifierNLTK\_l.show\_most\_informative\_features(260) dobijamo sledeći rezultat:

Most Informative Features

king = 1 shakes : modern = 41.0 : 1.0

fear = 1 shakes : modern = 30.6 : 1.0

going = 1 modern : shakes = 29.6 : 1.0

talent = 1 shakes : modern = 27.9 : 1.0

something = 1 modern : shakes = 24.1 : 1.0

looked = 1 modern : shakes = 21.5 : 1.0

mine = 1 shakes : modern = 20.8 : 1.0

peace = 1 shakes : modern = 20.5 : 1.0

fall = 1 shakes : modern = 19.8 : 1.0

soul = 1 shakes : modern = 18.5 : 1.0

set = 1 shakes : modern = 18.3 : 1.0

soldier = 1 shakes : modern = 17.8 : 1.0

son = 1 shakes : modern = 17.5 : 1.0

speak = 1 shakes : modern = 17.5 : 1.0

art = 1 shakes : modern = 17.1 : 1.0

show = 1 shakes : modern = 16.5 : 1.0

french = 1 shakes : modern = 15.8 : 1.0

brave = 1 shakes : modern = 15.8 : 1.0

horse = 1 shakes : modern = 15.8 : 1.0

eight = 1 modern : shakes = 15.5 : 1.0

stand = 1 shakes : modern = 15.5 : 1.0

sound = 1 shakes : modern = 15.1 : 1.0

fight = 1 shakes : modern = 14.7 : 1.0

power = 1 shakes : modern = 14.4 : 1.0

spirit = 1 shakes : modern = 13.8 : 1.0

body = 1 shakes : modern = 13.1 : 1.0

god = 1 shakes : modern = 13.1 : 1.0

master = 1 shakes : modern = 12.7 : 1.0

event = 1 shakes : modern = 11.9 : 1.0

die = 1 shakes : modern = 10.7 : 1.0

six = 1 modern : shakes = 10.5 : 1.0

sir = 1 shakes : modern = 10.5 : 1.0

tear = 1 shakes : modern = 10.4 : 1.0

alarm = 1 shakes : modern = 10.3 : 1.0

thousand = 1 shakes : modern = 10.3 : 1.0

turned = 1 modern : shakes = 10.1 : 1.0

men = 1 shakes : modern = 10.0 : 1.0

Pomoću ove naredbe možemo videti koje reči se najčešće sreću u Šekspirovim rečenicama a koje u modernim i na osnovu njih možemo praviti test primere u kojima klasifikator lako može pogrešiti.