Bogazici University
CmpE 493 Introduction to Information Retrieval, Spring 2018

Term Project

Turkish Sentiment Analysis for Twitter

Melike Ermiş - 2014400012
Hilal Benzer - 2013400027

May 22, 2018

Contents

1 Introduction **2**

2 Assumptions **2**

3 Program Structure **2**
3.1 Preprocessing 2
3.2 Classification 3

4 Usage **4**

5 Conclusion & Assesment **4**

6 References **6**

1 Introduction

In this project, we were asked to implement a sentiment analysis for Twitter, especially for university related tweets. We were provided a train set containing 1879 tweets with positive sentiment, 1121 tweets with negative sentiment and 3372 tweets with notr sentiment. What we need to do was perform preprocessing steps on training data, obtain data for classification, take test data as input and output their corresponding tags.

2 Assumptions

1. Required libraries / packages:
   - `Python 3`
   - `Numpy` for `Python 3`
   - `scikit-learn` for `Python 3`
2. Train documents in `src/Train` directory
3. Test documents in `tests`

3 Program Structure

3.1 Preprocessing

For processing raw material, we apply the steps below:

1. Lower all characters in raw text. (Case folding)
2. Replace "," and "" with whitespace because people tend to join two words with these characters.
3. Split the data using whitespace characters. (Tokenization)
4. Remove words starting with
   - `pic.twitter`
   - `@`
   - `rt`
5. Remove repeating characters, e.g. "günaaaaydıınn" is converted to "günaydın"
6. Replace emoticons. Before we got rid of punctuation, we decided to keep emoticons since they carry emotions and sentiment.
   - Replace ":)", ":-)", "=)", ":D", ":d", "<3", "(:", ":')", "^^", ";)", "(-:" with word "`SMILEYPOSITIVE`"
   - Replace ":-(", ":(", ";(", ";-(", "=(", ":/", ":\", "—_—", "):", ")-:" with word "`SMILEYNEGATIVE`"
7. Remove punctuation.
8. Correct spelling. In this part, Zargan Lexical Database is used to get the word forms and their frequencies.
9. Stem the words. In this part, the same database is used to get the word forms and their stems.
10. Remove words with length is less than or equal to 1.
11. Remove stopwords. Below is our stopword list:

- acaba
- adeta
- ait
- altı
- ama
- ancak
- artık
- aslında
- asıl
- ayrıca
- bazen
- başka
- belki
- ben
- beri
- beş
- bide
- bir
- biraz
- birkaç
- birçok
- biz
- bu
- bura
- böyle
- cuma
- cumartesi
- da
- dahil
- dair
- de
- defa
- diye
- diğer
- dokuz
- dolayı
- dört
- en
- et
- eğer
- fakat
- falan
- filan
- galiba
- gel
- gene
- gibi
- göre
- hadi
- hangi
- hem
- herhalde
- herhangi
- iki
- ile
- için
- kere
- kez
- kim
- kimi
- lakin
- lütfen
- mesela
- mi
- mü
- mı
- new
- niye
- ol
- on
- oysa
- pazar
- pazartesi
- pek
- perşembe
- rağmen
- resmen
- salı
- sekiz
- sen
- seni
- siz
- sırf
- tabi
- tabii
- tane
- the
- un
- vala
- var
- veya
- yada
- yahu
- yaklaşık
- yani
- yap
- yedi
- yoksa
- zaten
- çarşamba
- çok
- çünkü
- üzere
- üç
- şey
- şu

3.2 Classification

For comparison purposes, we decided to use different classification methods and different sizes of training sets. We also used these classification methods with *Bag of Words* method and *Bigram* method.

- **Multinomial Naive Bayes:** We tried packages `nltk` and `scikit-learn` for applying MNB classification and we settled for `scikit-learn` in the end.
- **Support Vector Machine:** We used `SGDClassifier` of `scikit-learn` library for this part.
- **K Nearest Neighbors:** We used `KNeighborsClassifier` of `scikit-learn` library for this part.

For different sizes of training set, we trained the classifier with full training sets and the first half of the training sets. There were times we obtained better results without removing the words in our stopword list, so we considered removing stopwords as a variable factor as well.

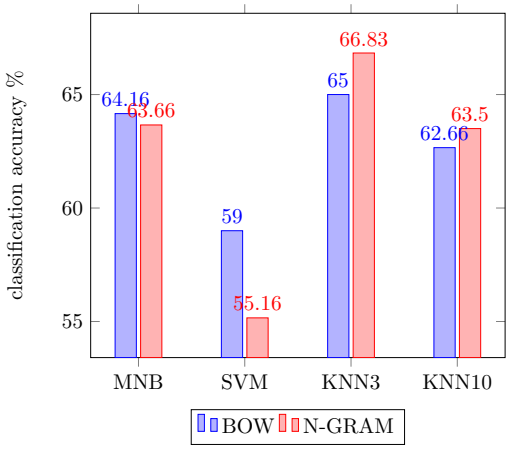
4 Usage

1. Pull the repository from GitHub.
2. `$ cd BOUNSentimentAnalysis/src`
3. To train `$ python3 Train_data.py`
4. To test using Bag of Words:
   - Multinomial Naive Bayes: `$ python3 Test_data_MNB.py`
   - Support Vector Machine: `$ python3 Test_data_SVM.py`
   - K-Nearest Neighbors: `$ python3 Test_data_KNN.py <neighbor_count>`
5. To test using 2-grams:
   - Multinomial Naive Bayes:`$ python3 Test_data_MNB.py bigram`
   - Support Vector Machine: `$ python3 Test_data_SVM.py bigram`
   - K-Nearest Neighbors: `$ python3 Test_data_KNN.py <neighbor_count> bigram`

5 Conclusion & Assesment

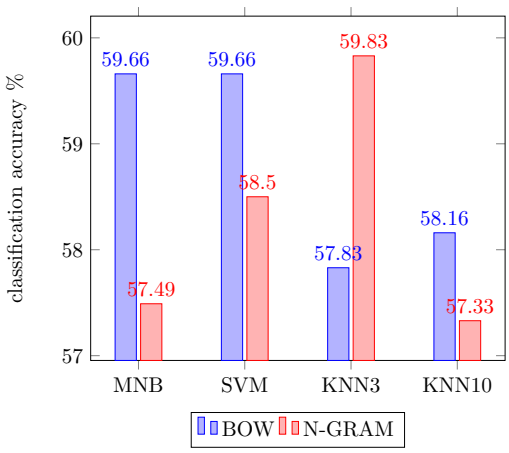
We tested our code with our data from part 1 of term project. We were asked to decide the sentiment of 600 tweets and send the data back. Our results with different classifications and methods are plotted below. According to the results, the most accurate classification method is

- Removing stopwords
- With full training test
- K nearest neighbor witk k being 3
- 2-gram



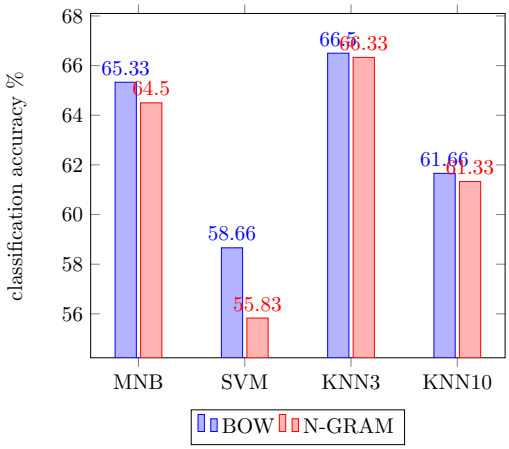
	MNB	SVM	KNN3	KNN10
BOW	64.16%	59.0%	65.0%	62.66%
BI-GRAM	63.66%	55.16%	66.83%	63.5%

Table 1: Full trained with removing stopwords



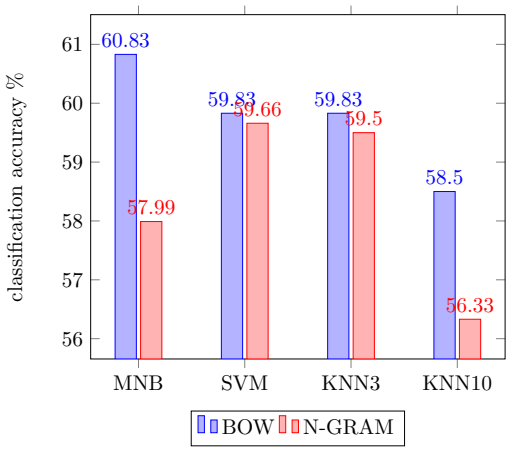
	MNB	SVM	KNN3	KNN10
BOW	59.66%	59.66%	57.83%	58.16%
BI-GRAM	57.49%	58.5%	59.83%	57.33%

Table 2: Half trained with removing stopwords



	MNB	SVM	KNN3	KNN10
BOW	65.33%	58.66%	66.5%	61.66%
BI-GRAM	64.5%	55.83%	66.33%	61.33%

Table 3: Full trained without removing stopwords



	MNB	SVM	KNN3	KNN10
BOW	60.83%	59.83%	59.83%	58.5%
BI-GRAM	57.99%	59.66%	59.5%	56.33%

Table 4: Half trained without removing stopwords

Positive	puan	itü	psikoloji	hedef	çocuk	ye	yüz	be	tıp
	sene	ora	lise	bitir	tercih	hukuk	teşekkür	düşün	hayal
	dil	tek	lan	güzel	yine	hayat			
Negative	rektör	ülke	kapa	yuva	yazık	baş	temizle	terör	amerikan
	akıl	yetiş	pk	hiç	işgal	el	hain	sev	vatan
	neden	gerek	hal	sosyoloji	devlet	terörist	sor	okuyan	bit
Notr	ödül	başkan	yürü	genç	konferans	devam	düzenle	etkin	program
	ekonomi	gerçek	hak	kat	araştır	taraf	güney	tarih	grup
	eğitim	üye	kuzey	hafta	kadın	buluş	kulüp	konuş	zirve

Table 5: Most informative words with the classification methods we used

6 References

1. Zargan Lexical Database
2. scikit-learn for Python
3. Türkçe Twitter Mesajlarının Duygu Analizi Sentiment Analysis for Turkish Twitter Feeds