Akın Toksan
2010400186

CMPE561 – Report for Homework 1

GitHub repository: https://github.com/akintoksan/cmpe561_project1

1) For tokenization, I just trimmed the first and last characters (recursively) and apostrophe. (e.g. "Ankara'da…" → "Ankara") This also eliminates the Turkish characters because I couldn't handle the encoding but the results are good so I guess it is not that important overall.

I read all the documents token by token and put the normalized token (word) to the lexicon if it was not in the lexicon before or updated the row of the word. So for each word I had a row of 69 (number of authors) to keep the frequencies. It can be thought as a 55K x 69 matrix. (55072 is the size of the lexicon for the training set that I lastly created)

Number of documents for the training set

abbasGuclu: 9
abdullahAymaz: 6
ahmetAltan: 6
ahmetHakan: 9
aliBulac: 6
atillaDorsay: 9
ayseArman: 9
balcicekPamir: 18
bekirCoskun: 6
bulentKorucu: 6
canDundar: 6
cemilErtem: 6
cemSuer: 6
cengizCandar: 6
cetinAltan: 18
deryaSazak: 6
doganHizlan: 9
eceTemelkuran: 6
ekremDumanli: 18
elifSafak: 6
emreAkoz: 18
emreKongar: 9
ergunBabahan: 6
ertugrulOzkok: 6
fatihAltayli: 18
fehmiKoru: 6
fikretBila: 6
fundaOzkan: 6
gulseBirsel: 6
guneriCivaoglu: 6
hakkiDevrim: 9
hasanCemal: 6
hasanPulur: 6
hasmetBabaoglu: 6
hekimogluIsmail: 6
hincalUluc: 6
huseyinGulerce: 6
ismetBerkan: 6
mahfiEgilmez: 9
mehmetaliBirand: 6
mehmetBarlas: 18
mehmetOz: 9
mehmetTezkan: 6
melihAsik: 6
mumtazerTurkone: 6
muratBardakci: 6
muratBelge: 6
muratYetkin: 6
nazliIlicak: 6
nihalKaraca: 6
nurayMert: 6
omerUrundul: 6
oralCalislar: 6
raufTamer: 6
rehaMuhtar: 6
ridvanDilmen: 9
ruhatMengi: 9
samikohen: 18
savasAy: 6
serpilYilmaz: 6
tahaAkyol: 18
tamerKorkmaz: 6
tarhanErdem: 6
umurTalu: 6
yaseminCongar: 6
yigitBulut: 6
yilmazOzdil: 6
yukselAytug: 9
zekiCol: 6

Number of documents for the test set

abbasGuclu: 6
abdullahAymaz: 4
ahmetAltan: 4
ahmetHakan: 6
aliBulac: 4
atillaDorsay: 6
ayseArman: 6
balcicekPamir: 12
bekirCoskun: 4
bulentKorucu: 4
canDundar: 4
cemilErtem: 4
cemSuer: 4
cengizCandar: 4
cetinAltan: 12
deryaSazak: 4
doganHizlan: 6
eceTemelkuran: 4
ekremDumanli: 12
elifSafak: 4
emreAkoz: 12
emreKongar: 6
ergunBabahan: 4
ertugrulOzkok: 4
fatihAltayli: 12
fehmiKoru: 4
fikretBila: 4
fundaOzkan: 4
gulseBirsel: 4
guneriCivaoglu: 4
hakkiDevrim: 6
hasanCemal: 4
hasanPulur: 4
hasmetBabaoglu: 4
hekimogluIsmail: 4
hincalUluc: 4
huseyinGulerce: 4
ismetBerkan: 4
mahfiEgilmez: 6
mehmetaliBirand: 4
mehmetBarlas: 12
mehmetOz: 6
mehmetTezkan: 4
melihAsik: 4

mumtazerTurkone: 4	omerUrundul: 4	savasAy: 4	yigitBulut: 4
muratBardakci: 4	oralCalislar: 4	serpilYilmaz: 4	yilmazOzdil: 4
muratBelge: 4	raufTamer: 4	tahaAkyol: 12	yukselAytug: 6
muratYetkin: 4	rehaMuhtar: 4	tamerKorkmaz: 4	zekiCol: 4
nazliIlicak: 4	ridvanDilmen: 6	tarhanErdem: 4	
nihalKaraca: 4	ruhatMengi: 6	umurTalu: 4	
nurayMert: 4	samikohen: 12	yaseminCongar: 4	

2) Naïve Bayes

a) BoW Only

For the Bag of Words approach, I kept total term frequencies for each word in lexicon. Then I calculated a *dividend* and a *denominator* which dividend keeps the value of a word's total appearance in a class (for Laplace smoothing I added alpha of 0.025 to the dividends). Denominator is calculated by summing total number of words appeared in a class and lexicon size multiplied with alpha. The finally after calculating the values for each word, a summed their logs for each class. I also added the probability of the class to this result.

Using only the BoW approach and not any other features I found the values as below:

```
Macro Average Precision: 0.31884057971
Macro Average Recall: 0.159420289855
Macro Average F-score: 0.212560386473

Micro Average Precision: 0.593406593407
Micro Average Recall: 0.593406593407
Micro Average F-score: 0.593406593407
```

b) BoW + Average Word Length

For average word length approach I first calculated the average word length of each document in the training set. Then I calculated the mean and variance for each class separately. After the I calculated the probability density function (pdf) of a document and added its log to the BoW probability. I didn't give equal weight to BoW and Average Word Length (AWL) because average word length of each class appears to be very close to each other (the length of between 6 and 7) So I weighted it as;

$$0.8*BoW + 0.2*AWL$$

Because if we gave them equal weights the result is as below:

```
Macro Average Precision: 0.304347826087
Macro Average Recall: 0.144927536232
Macro Average F-score: 0.196353436185

Micro Average Precision: 0.576923076923
Micro Average Recall: 0.576923076923
Micro Average F-score: 0.576923076923
```

So as we can see, all the results get lower than the BoW Only approach. When we

weight them all the values get to 1. So it classifies all the documents correctly.

b) BoW + Average Number of Words (ANW)
As we can see the results below, unweighted BoW + ANW approach works a little bit worse than the previous approach.

```
Macro Average Precision: 0.289855072464
Macro Average Recall: 0.144927536232
Macro Average F-score: 0.193236714976

Micro Average Precision: 0.57967032967
Micro Average Recall: 0.57967032967
Micro Average F-score: 0.57967032967
```

Weighted approach also gives all the values to 1. (0.8*BoW + 0.2*ANW)

c) BoW + AWL + ANW

Unweighted:

```
Macro Average Precision: 0.275362318841
Macro Average Recall: 0.144927536232
Macro Average F-score: 0.189905047476

Micro Average Precision: 0.576923076923
Micro Average Recall: 0.576923076923
Micro Average F-score: 0.576923076923
```

As you can expect, all the precision, recall and f-scores are also 1 in this approach.

d) Analysis for the approaches

BoW worked better alone instead of combining the Naïve Bayes with other feature sets (if we do not weight them).

I actually expected the other way, but when a weighted the probabilities due to their priorities it, my program worked perfect. But without weighting them adding extra features lower the accuracy of the system somehow.

---

ps. Program only gives the result of BoW + 2 features. If you want to test the approaches separately, others are commented out between the lines 207 – 214. Don't forget to comment out the line 205 also, if you want to run the program with other approaches.

ps2. It is assumed that the training set is not too small. In that case variance of a class might be zero, which gives an error on my code because of the formulas makin the denominator value zero:

tmp_base = 1/sqrt(2*pi*tmp_var)
and
tmp_pow = -((avg_len-tmp_mean)2)/(2*tmp_var)**