MSiA490 Text Analytics hw2 write-up

Xiang Shawn Li

**Q1: For each line where the tokenization differs between the two approaches, show the original string and the two alternative tokenizations. What patterns do you see in the differences?**

Major differences are highlighted out in the excel sheet containing all results.

Based on the comparison, lucene tokenization removed all non-number and non-letter characters first and then tokenizing, but CoreNLP saves hyphenated words and separates all different type of characters into new tokens.

For example, original text “$70-a-share” is changed to “70 a share” 3 tokens by Lucene but kept major format “70-a-share” by CoreNLP. Same example can be “buy-back”. For those special words, it is misleading if they are separated into different tokens. CoreNLP does a smarter choice on it.

CoreNLP tokenizes each different type character into a new token but Lucene just simply deletes all non-number and non-letter characters, which do matter. For example, dollar sign, “$”, which is removed by Lucene and saved by CoreNLP, define better the following number.

Overall, for the tokenization, CoreNLP does better job than Lucene.

**Q2: Describe the main ways the lemmatizer differs from the stemmers. Also describe the patterns you see in the differences between stemmers.**

Major differences are highlighted out in the excel sheet containing all results.

All normalization functions from Lucene are doing stemming, which simply cut words into their base or root form. CoreNLP lemmatizer converts words to simplest form by involving complex tasks such as understanding context and determining the part of speech of a word in a sentence. (Wikipedia)

Word stemming works fast since it does not require any further steps of words analysis. However, results from stemming are not precise and a little bit confusing. Lemmatizer works better store original words if necessary with analysis. Moreover, CoreNLP detects names of company, people, month, weekday and ect. and stores those words in original format. It helps CoreNLP to win the game here.

**Q3:**

See output files

**Q4: Basic frequency analysis**

```
[('datum', 127),
 ('analytic', 104),
 ('text', 82),
 ('work', 69),
 ('program', 43),
 ('University', 42),
 ('project', 41),
 ('use', 39),
 ('research', 36),
 ('field', 35),
 ('analysis', 35),
 ('year', 32),
 ('interested', 29),
 ('business', 29),
 ('interest', 27),
 ('Northwestern', 27),
 ('science', 24),
 ('develop', 24),
 ('degree', 24),
 ('graduate', 24)]
```

The list gives information about this corpus and backgrounds of the class. For example, top two words with the most frequency is ‘datum’, ‘analytics’ and ‘analytics’. It tells major people of the class talk about analytic and data and gives information about their professional background. Words ‘northwestern’, ‘university’ inform the class is from Northwestern University.

**Q5: Basic bigram analysis**

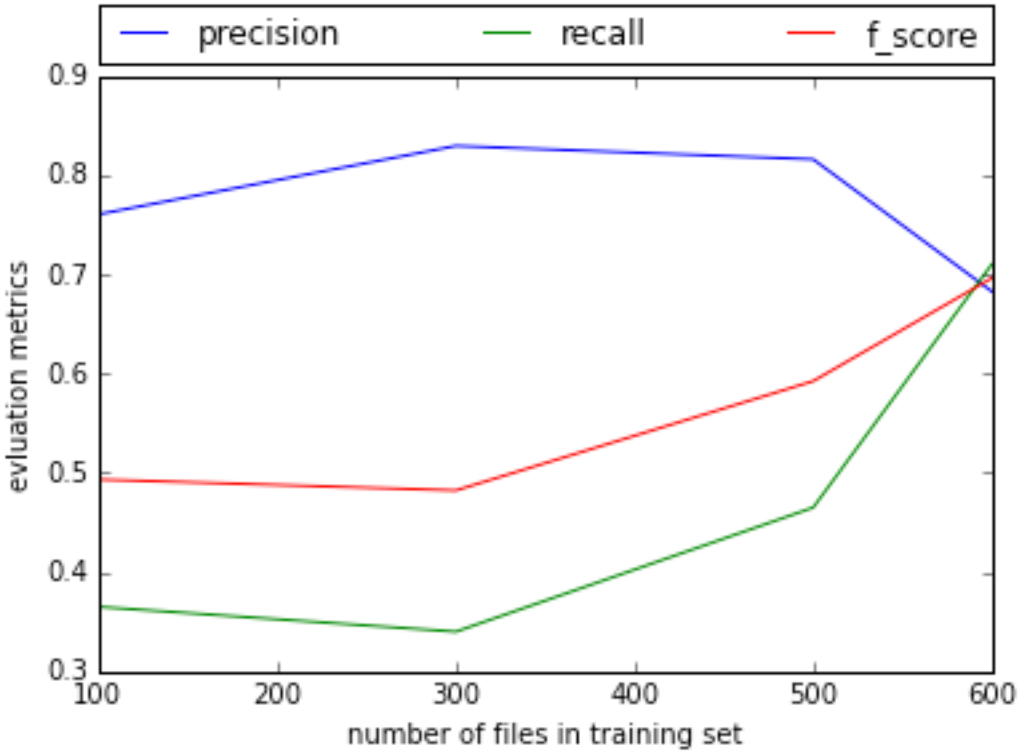
```
[('text analytic', 46),
 ('Northwestern University', 17),
 ('datum analysis', 14),
 ('msium program', 13),
 ('become interested', 12),
 ('datum science', 12),
 ('machine learning', 11),
 ('text datum', 8),
 ('datum scientist', 7),
 ('text mining', 6),
 ('computer science', 6),
 ('big datum', 6),
 ('social media', 6),
 ('look forward', 6),
 ('unstructured datum', 6),
 ('undergraduate degree', 5),
 ('predictive model', 5),
 ('datum analytic', 5),
 ('also work', 5),
 ('datum mining', 5),
 ('would like', 5)]
```

The list gives information about this corpus and backgrounds of the class. For example, top bigram with the most frequency is ‘text analytic’, which tells the main topic of the corpus is about text analytic. ‘Northwestern University’ is information of the class. ‘datum science’, ‘datum scientist’, ‘big datum’ and ‘machine learning’ tells the main focus of people in the class is data science.

**Q6: Sentiment analysis**

Test results of cv6 and cv7 by different models: (F score’s beta = 0.5)

```
          Precision            ,  Recall               ,  F score
100 files 0.76041666666666663,  0.36499999999999999,  0.4932432432432432
300 files 0.82926829268292679,  0.34000000000000002,  0.48226950354609938
500 files 0.81578947368421051,  0.46500000000000002,  0.59235668789808926
600 files 0.68269230769230771,  0.70999999999999996,  0.69607843137254899
```



There is a trade-off between precision and recall as the graph shown. F-score is a mean of these two measurement. Therefore, a model with a high F-score can perform better. Therefore, based on the graph of results, it shows performance would improve with more examples.

Performance of final evaluation on cv8 to cv9:

Precision , Recall , F score
0.61693548387096775, 0.76500000000000001, 0.68303571428571441

The prediction output is included.