Assignment – 3B

Prerak Patel - 000825410

# **Data Description**

**Data set name:** Reuters Corpus 21578

**Source:** https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

**Description of Classification Task:**

There are 5 binary classification tasks chosen to test against different algorithms.

1. Earn: All the news article containing “Earn” in the topic section are classified as Earn to separate between “Earn” articles and “NOT Earn” articles. There are 3987 “Earn” articles in the corpus.
2. Acq: All the news articles containing news related to corporate acquisition are classified as “Acq” and there are 2448 articles in the corpus.
3. Money-fx: All the news articles related to foreign exchange money are classified as “money-fx” and there are 801 articles in the corpus.
4. Crude: All the news articles related to crude oil are classified as “crude” and there are 634 articles in the corpus.
5. Grain: All the news articles related to grain are classified as “grain” and there are 628 articles in the corpus.

## **Training and Testing Split:**

We needed to compare the performance of different algorithms and for that we need the exact same training and testing data to check the accuracy, precision and recall for each version. So, we have separated first 80% of the dataset for training and other 20% of the dataset for testing purposes. And used the same dataset throughout all the version to carry out the calculations.

**Statistics:**

Number of features:

There is just one feature for each run as there no columns to these dataset once we read the corpus and convert that in bag of words we just have 2-d array with long string of the numbers which represents how many times that letter has occurred respected to the vocabulary list.

Number of items:

We have read first 500 articles for the programming purposes so all the results shown below will be shown using those results, that means there are 500 items in each class.

Number of items in training and testing split:

As we mentioned earlier, to compute the results we have separated data into 80/20 split. So, we have 500 articles out of which 80% is used to training split which comes out to be 400 articles and remaining 100 articles which is other 20% is used for testing purposes.

# **Description of Text Representation**

For all the different text representation mentioned below we have created the dataset every time and then separated those into training and testing split to use. Thereafter, we have looped through five task labels to run each text representation.

1. Bag of Words w/ Multinomial and Complement

* We have used “bagOfWords” function designed by “Sam Scott” in text\_classification.py. “Bag of Words” function expects two arguments first vocabulary list which consists of all the words occurring in the dataset. And second argument article that we want to classify which consists of list of words occurred in that article. Inside bag of words function it used dictionary to store each unique word with its counter value stored. First loop will iterate through the words in the article and will be added to the dictionary if the word is not present in the dictionary but if it is present in the dictionary it will increase occurrence value by “1”. Second loop will iterate through the vocabulary list and it would append occurrence value of the word to the bagOfWords array if the vocabulary word exists in the dictionary and if it does not it will just put 0 on that index. That bagOfWords array will then be used to predict the testing articles using Multinomial NB and complement NB to produce results.

1. Bag of Stems w/ Multinomial and Complement

* “stemWords” function expects list that it will stemmed to the root word. It will be stemmed from list of

["exchangeables", "works", "jobs", "sliced", "diced", "dooly's", "driving", "driving", "picking", "fractured", "fracturer", "refracture", "frosts"," unable"] to

['exchangeable', 'work', 'job', 'slice', 'dice', 'dooly', 'driv', 'picking', 'fracture', 'frost', 'able']

* It stems words with suffix of s, es, er, ed, ly, ing and ‘s and prefix of re, de, un, dis and mis. Function will return stemmed list which will then be used to predict the testing articles using Multinomial NB and complement NB to produce results.

1. Bag of Words w/ Stop Words w/ Multinomial and Complement

* “stopWords” function is implanted using nltk library. It has stop words set for English included in it. So, the stopWords function expects a list, which we will iterate through each word in the list and check if it is a stop word and if it is, we will not add that into stopWords array. This will return a list which eradicates words like “the”, ”is”, “or”, “between”, and many more. We will pass vocabulary list and each article document to the stopWords function before passing it to calculate bagOfWords. Now, we will have bagOfWords much shorter than in text representation 1, we will now use this to predict the testing articles using Multinomial NB and complement NB to produce results.

1. Bag of Stems w/ Stop Words w/ Multinomial and Complement

* Here we are going to pass vocabulary list and each article first through “stopWords” function and resulted array would then be passed to “stemWords” function this will return array with all the stop words removed and words being stemmed. Now we will use this array to predict the testing articles using Multinomial NB and complement NB to produce results.

1. Set of Words w/ Bernoulli NB

* Here we are going to use binary representation for the word occurrences in each article, before we use bag of word that will track the number of occurrences of each word. Instead, set of words will just check if the word has occurred or not and if it has occurred in the article it will just put 1 in the array and 0 if it does not appear in the article. Now we will use this array binary array to predict testing articles using Bernoulli NB to produce results.

# **Results**

Below we shown screenshots of the each run with the limit of articles.

1. Bag of Words w/ Multinomial and Complement

Table

Description automatically generated

1. Bag of Stems w/ Multinomial and Complement

Text, letter

Description automatically generated

1. Bag of Words w/ Stop Words w/ Multinomial and Complement

Table

Description automatically generated

1. Bag of Stems w/ Stop Words w/ Multinomial and Complement

Table

Description automatically generated

1. Set of Words w/ Bernoulli

A picture containing table

Description automatically generated

# **Discussion**

1. **Are there clear winners or losers?**

* There were no clear winners or losers as the precision and recall for all of them lies between 0.75-0.85 and 0.54-0.65 respectively. There is also not a large spread in accuracy between different text representations.

1. **Give some solid idea for why some text representations might be better or not better than others?**

* Set of words produced great accuracy but performed poorly in recall as it just scored “0” meaning the article does not belong to that topic most of the time. So, it was able to produce more accuracy but poor recall. Bag of stems w/ stop words performed really well and produced great results as the data set passed was smaller comparatively and it was able give some decent results in precision and recall. So, I believe set of words was not better performer than other whilst Bag of stems w/ stop words was best among all.

1. **Do the results make sense to you? Why or why not?**

* Results from all the text representations made sense to me as we made changes to the data set, text representations was able to generalize the text more and produced better results precision wise, but accuracy remained almost the same. Bag of words w/ stop words w/ Multinomial and Complement and Bag of stems w/ stop words w/ Multinomial and Complement produced the exact same results, which does not make sense as we are making the dataset even shorter and more effective by stopping words and then stemming it.

# **Future Work**

If you had more time, where would you go next? What other variations of text representation would you like to explore? What other algorithms or data sets would you like to use? What other tests would you like to do?

1. **If you had more time, where would you go next?**

* If I had more time, I would try this text representations on different datasets from UCI.

1. **What other variations of text representation would you like to explore?**

* If I had more time to explore, I will try tf-idf with stemming, stopping words and removing most frequent words that does not hold meaning as one whole text representation and check how does it performs. I would also try other permutations tf-idf to find the best text representation.

1. **What other algorithms or data sets would you like to use?**

* I would like to use social media data set to classify it as good post or bad post. I will also explore more to find datasets related to movies to classify which movie is it from the captions body.

1. **What other tests would you like to do?**

* I don’t have any other tests in my mind that we will give me better understanding than confusion matrix, precision, recall and accuracy.