CSE 472 - SOCIAL MEDIA MINING
PROJECT II REPORT
FAKE NEWS CLASSIFIER

SUBMITTED TO:
Dr. HUAN LIU

BY
NEEL PATEL -1210329585
SUBRAMANYA BHASKARA TARUN LOLLA - 1216095647

Table of Contents

COVER PAGE 0

INTRODUCTION 2

PRE-PROCESSING AND FEATURE ENGINEERING 2

MODEL DESCRIPTION 2

RESULTS 3

IMPLEMENTATION 3

REFERENCES 4

Introduction

The task given to us for this project was to build a fake news classifier. As mentioned in the Project instructions document, social media is now the main source of news for many people and since spreading misinformation and fake news on social media is relatively cheaper and easier to spread a lot of organizations and people try to spread fake news and misinformation for many reasons such as financial or political gain. This can have a negative impact on our society mainly because it leads to people having biased views or people losing faith in the credibility of news report they read. Hence, it is important to detect and classify fake news to prevent spread of misinformation.

Pre-processing and Feature Engineering

Pre-processing:
To clean the data and obtain better results we decided to ignore certain lines in the training data that had comma in it which lead to an extra column being added. Such lines were few in number hence we decided to ignore them.

Feature Engineering:
We only used the English title for our training and model because it was hard to interpret the Chinese titles.
Word count/frequency, Term Frequency and Inverse Document Frequecny (TF-IDF) were used to extract features from the text and assign weights to them.

Model Description

We used the multinomial Naïve Bayes Classifier model for our approach.
First, we used a CountVectorizer to count the frequency of words in the titles and tokenize them and then we used a TF-IDFVectorizer to help us find the weights of words based on their frequency and how they appear in the titles.
Then we used the multinomial Naïve Bayes Classifier to train our model with the matrix obtained from the TFIDF matrix to classify our titles and assign them the value of ‘agree’, ‘disagree’ or ‘unrelated’.
A brief explanation of multinomial Naïve Bayes Classifier is that it helps us estimate the conditional probability of feature relative to a class and helps us classify the feature based on the frequency of that feature appearing for that class in the training data.

Results

Based on our training model and classification we were able to obtain an **accuracy of 74.42%** in classifying the news articles.

The same results are printed out to the end of the execution of the script, fakeNewsClassifier.py

Below you can find the Classification report that provides the Precision, Recall and F-scores:

	Precision	Recall	F1-score	Support
agree	0.78	0.29	0.42	18567
disagree	1.00	0.01	0.02	1684
unrelated	0.74	0.96	0.84	43836
Micro Average	0.74	0.74	0.74	64087
Macro Average	0.84	0.42	0.43	64087
Weighted Average	0.76	0.74	0.70	64087

Implementation

The project is setup in Python. The user is expected to install Python along with libraries numpy, pandas, matplotlib, scikit-learn to execute the scripts in this project.

In addition to this, the user must change the content of the "dataset.txt" file. The user should replace the file path of each file with the file path on their local computer. Please note that there shouldn't be any spaces in each of the three lines.

For example,

```
train_data=/Users/Neel/PycharmProjects/cse472-smm-project2/Project2_Files/train.csv
validation_data=/Users/Neel/PycharmProjects/cse472-smm-project2/Project2_Files/validataion.csv
test_data=/Users/Neel/PycharmProjects/cse472-smm-project2/Project2_Files/test.csv
```

On completion of the above, run the file, fakeNewsClasssifier.py prints the accuracy metrics and also outputs the file submission.csv to the same directory.

References

We used the below references to complete this project.

- Naive Bayes algorithm in Machine learning Program | Text Classification python (2018). Youtube (2019). https://www.youtube.com/watch?v=0kPRaYSgblM&list=LLnQe8zSv_LO_pGEQxJGVScQ&index=3&t=0s
- https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf