Political Ideology Detection on Twitter

Anonymous NAACL submission

Abstract

Political Ideology detection is one of the major concerns of the modern era as it plays a huge role in deciding the election results. In current era, data is considered as one of the prized possession of a human which is leveraged by organizations to run decisions in their favour. In this paper we propose a method to analyze a user's Twitter timeline and classify him as right or left leaning.

1 Introduction

Everyone is aware of the 2016 US presidential elections. Trumps rise to power is heavily dedicated to his data analysis team which helped him identify important topics in particular groups across the states to address about which indeed helped him win the elections.

Prior work has been done in analyzing the tweets and gaining useful insights (Conover et al., 2011a). Other than that data from Twitter has been used to analyze various trending topics or public figures. In this paper we devise a method to classify a user into left leaning or right leaning using the publicly available Twitter data. As per our knowledge no such work has been done in Literature as of now.

2 Data and Methods

2.1 Twitter Lists

We have collected data from the user handles available on the Twitter Handle of Media Bias Fact Check(MBFC) where they have created two separate lists for Right Leaning and Left Leaning. We collected the data from the these Twitter handles and considering the ground truth provided by MBFC. The Twitter Lists consisted of 368 left leaning twitter ids and 164 right leaning twitter ids.

Later, we also collected data from twitter handles of certain sources which were listed on the MBFC website. From there we obtained 109 right leaning ids and 86 left leaning ids. Therefore, In total we had 477 left leaning ids and 250 right leaning ids.

As per the guidelines of Twitter we can extract 3200 latest tweets from these twitter handles. We have collected the tweets using Tweepy which is an official package for collecting data from twitter.

2.2 Public Dataset

In (Gu et al., 2016; Sumit Bhatia, 2018) proposed a public dataset containing of user twitter ids which were labelled into 7 categories i.e. extremely left leaning to extremely right leaning. We were able to get the user ids using which we collected the data. We have used this as the test dataset and used the annotation provided in (Conover et al., 2011b) as ground truth.

We have used one more publicly available dataset which consists of 2 million tweets without annotation. We plan to use a subsample the tweets and try to manually annotate. Moreover, this would add as a separate experiment where we would classify the user on the basis of a single tweet.

2.3 Methodology

The tool used for the classification of users into left or right leaning is MALLET. MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text(http://mallet.cs.umass.edu/). It includes sophisticated tools for document classification: efficient routines for converting text to "features", a wide variety of algorithms (including Nave Bayes, Maximum Entropy, and Decision Trees), and code for evaluating classifier performance using several commonly used metrics. In addition to sophisticated Machine Learning applications, MALLET includes routines for transforming text documents into numerical representations that can then be processed efficiently. This process is implemented through a flexible system of ”pipes”, which handle distinct tasks such as tokenizing strings, removing stopwords, and converting sequences into count vectors.

The raw data collected from the MBFC twitter handle is processed according to the format which the MALLET tools take as an input. It represents data as lists of ”instances”. All MALLET instances include a data object. An instance can also include a name and (in classification contexts) a label. For example, if the application is guessing the label of Twitter users ideology as left or right, an instance might consist of a vector of word counts (data), the user_id of the Twitter user (name) and the label of the page as 0 for left and 1 for right (label). There are two primary methods for importing data into MALLET format, first when the source data consists of many separate files, and second when the data is contained in a single file, with one instance per line. We used the latter method.

One file, one instance per line: The data is in the following format: [user_id] [label] [text of the tweet...]. In this case, the first token of each line (whitespace delimited, with an optional comma) becomes the instance name, the second token becomes the label, and all additional text on the line is interpreted as a sequence of word tokens. Note that the data, in this case, will be a vector of feature/value pairs, such that a feature consists of a distinct word type and the value is the number of times that word occurs in the text. The data is converted into .mallet format which is fed into the classifier for further classification. The data is also cleaned by removing the stop words and converting all the text into the lower-case(MALLET by default converts all word features to lowercase). Following command is used for the task: *bin/mallet import-file –input /data/web/data.txt –output web.mallet –remove-stopwords.*

Document Classification: A classifier is an algorithm that distinguishes between a fixed set of classes, such as ”left” vs. ”right”, based on labeled training examples. MALLET includes implementations of several classification algorithms, including Nave Bayes, Maximum Entropy, and Decision Trees. In addition, MALLET provides tools for evaluating classifiers. The training data prepared is used to train the classifier using the *bin/mallet train-classifier –input training.mallet –output-classifier my.classifier –training-portion 0.8 –trainer Naivebayes –cross-validation 10* which takes the parameters for input training data, output classifier, the split ratio for splitting the data for training and testing the classifier, a training algorithm and number of folds for cross-validation.

Table 1: Data Distribution

Data	Left	Right
User Ids	477	250
Tweets	0.26 M	0.1 M

3 Results

We have summarized the results of our experiments in Table 2. We have reported the macro avg scores for Precision, Recall and F1 Score.

We have preprocessed the data by removing stop words and convertin g them to lowercase and generated a bag of words model before feeding the data to the classifier. We have used 5 fold cross validation for reporting the results.

From the results it is observed that Naive Bayes consistently performs well compared to the rest two classifiers i.e. Decision Tree and Max Entropy which can be attributed to the fact that there lacks the usage of semantic hierarchy for classifying a user. It can be said that usage of some words sequentially in the tweet would trigger to classify the user to be right or left leaning.

According to our analysis due to the usage multiple contrasting words sequentially would be confusing the Decision Tree and Max Ent classifier which would be leading to wrong decisions.

Table 2: Results (%)

Classifiers	Naive Bayes	MaxEnt	Decision Tree
Accuracy	70	34	67
Precision	67	67	58
Recall	67	50	59
F1 score	67	25	57

4 Further Work

We plan to extend our work by pretraining the classifier with one of the datasets and testing them on a completely different dataset as that would provide a better overview about the classifier and it's scalability. We also plan to incorporate the use of sentiment and hashtags as an input to the classifier based on the hypothesis that sentiment would remain universal across most of the topics and collaborating hashtag with sentiment would allow to see the support of the user to a particular topic and indeed help in classifying the user into left or right leaning.

References

M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer. 2011a. Predicting the political alignment of twitter users. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 192–199.

Michael Conover, Jacob Ratkiewicz, Matthew R. Francisco, Bruno Gonalves, Filippo Menczer, and Alessandro Flammini. 2011b. Political polarization on twitter. In *ICWSM*.

Yupeng Gu, Ting Chen, Yizhou Sun, and Bingyu Wang. 2016. Ideology detection for twitter users with heterogeneous types of links. *CoRR*, abs/1612.08207.

Deepak P Sumit Bhatia. 2018. Topic-specific sentiment analysis can help identify political ideology. In *EMNLP*.

5 Supplemental Material

Link: https://goo.gl/3tH4Bi