CS-583 Data Mining and Text Mining Project
TWITTER SENTIMENT ANALYSIS

Rochana Chaturvedi
UIN: 662511080
rchatu2@uic.edu

Rishabh Goel
UIN: 653865205
rgoel20@uic.edu

Department of Computer Science
University of Illinois, Chicago

ABSTRACT

Twitter is a popular social network for microblogging. It allows people to express their opinions through short text snippets called tweets on a myriad range of topics. In this project, we perform sentiment analysis on tweets related to the 2012 United States presidential election between Barack Obama and Mitt Romney. The tweets are labelled with the candidate information and classified under various sentiment categories including positive, negative, and neutral. We apply several machine learning models to automatically infer the sentiments conveyed by a tweet. We find that a fine-tuned version of state-of-the-art model RoBERTa for sentiment classification achieves the best F1 score of 70.27% averaged across 10-fold cross-validation.

1. INTRODUCTION

Every day, Twitter sees millions of tweets posted by the public expressing their opinion on a wide range of topics. With the advances in natural language processing and machine learning techniques, it is now possible to analyse public sentiment at this scale for applications varying from stock market fluctuations to election outcome prediction.

In this project, we focus on the tweets collected during United States presidential election of 2012 mentioning either of the presidential candidates Barack Obama or Mitt Romney. Our objective is to train a classifier that can automatically label a tweet as carrying either positive, negative, or neutral sentiment with high accuracy.

As twitter is a social networking site, the language use is not bound by formal restrictions. A tweet often contains internet acronyms, spelling mistakes, emoticons, non-english characters, user references, web URLs etc. We pre-process each tweet to reduce the noise. Pre-processing is an important step to improve the model training. Further, the labelled training data can be skewed in terms of the sentiment expressed. Hence, we take care of the weight imbalance during training if required. We try two classes of models—bag of word (BoW) models which do not consider either the context or the order of words to be important and transformer-based large language models trained on large amount of text using transfer learning. We evaluate all the models using stratified 10-fold cross-validation with same splits for all models. Our fine-tuned version of RoBERTa sentiment classifier gives the best average accuracy 71.12% and average F_1 score 70.27% which is at least 12% more than our best BoW model.

We provide details about the data and pre-processing in Section 2 and describe our methods in detail in Section 3. We present results of experimental evaluation in Section 4 and summarize our conclusions in Section 5.

2. DATA

The data contains two files Obama.csv and Romney.csv. As the name implies, each file contains tweets expressing user's opinion on one of the two candidates. Each file contains the tweet, date and time of posting and sentiment annotated by human annotator. There are 7197 Obama-specific tweets and 7200 Romney-specific tweets. Each file contains several sentiment categories with following encoding— -1, 0, 1, 2, irrelevant, irrelevant, IR, and !!!!. Here -1 refers to negative sentiment, 0 to neutral and 1 to positive and 2 to mixed while others are noisy labels.

2.1 Pre-processing

As the first step, we filter observations with tweets containing nan or empty character, observations with labels other than -1, 0 or 1 and we also drop duplicate rows. The final statistics are reported in Table 1. We can get some idea of the election outcome by looking at these statistics if we treat this sample as representative of the population, and if we assume that on an average, the users favouring each party have same frequency of tweets. There are more tweets expressing negative sentiment towards Romney and more tweets expressing positive sentiment towards Obama. There is also a better incentive for Obama to sway neutral voters in his favor by studying them further.

Candidate	Positive	Neutral	Negative
Obama	1678	1976	1964
Romney	1075	1680	2892
Combined	2753	3657	4856

Table 1: Data Statistics

2.1.1 Text Pre-processing

As stated previously, the tweets are informal posts inundated with noise. We need to pre-process them before training machine learning classifiers, so they are not biased by noise. We perform following pre-processing steps using regular expressions:

- Usernames like @johnDoe were replaced with @user
- Hyperlinks beginning with http:// were replaced with http

The tokenizers of the transformer-based large language models are trained on noisy tweet text. Therefore, we do not have to perform further pre-processing for them. In addition, the candidate named entities are marked by text span <e> and </e> and hyperlink texts by <a> and . We observe from experiment on data subsample that the language models are benefitting from this extra information therefore we retain these as well. However, our Bag-of-Word models are not exposed to large tweet data apriori and will be learning from the limited training data at hand. Therefore, we perform following additional pre-processing for robustness[1]

- We remove '<a>', '', '<e>' and </e> tags
- The tweets are converted to lower case
- Tweet hashtags are removed. E.g. #voteForObama
- All mentions beginning with @ are removed including @user from previous step
- All urls are removed, including http from previous step
- All punctuations are removed including '','"",!,?,;,% etc
- All digits are removed
- Finally, extra white spaces are removed

3. Methodology

3.1 Bag of Words (BoW) Models

In this set of models, the text is represented as a bag of words disregarding the word order. This model is commonly used for document classification where the frequency of each word is used as a feature for training a classifier. We experiment with both word-ngram as well as character-ngram

[1] We also experimented with stemming and stopword-removal. However, it lowered the performance. We also considered leveraging emoticons but they are rare in the data.

models. The ngram is a continuous sequence of n items (words or characters). Each bag can contain binary 1-hot representation for each term (ngram) or numeric frequency-based representation of each term. We choose normalized frequency-based representation of TF-IDF vectors**.**

Tfidf Vectorization This representation accounts for the term frequency in each document to represent which term might be important for each class, and also the inverse document frequency so that common terms across all documents are penalized as they do not help in distinguishing the documents. In the present context, each tweet is a document. The TF-IDF formulation is as follows:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

$$TF(t,d) = \frac{N_{t,d}}{N_d} \quad \text{and,}$$

$$IDF(t) = ln\frac{1+n}{1+DF(t)} + 1$$

Where $N_{t,d}$ is the frequency of term t in tweet d. N_d is total number of terms in d. DF(t) is the number of tweets containing term t, and n is the number of tweets in the training data.

Models

We choose a Logistic Regression (LR) classifier[2]. LR models the probability of a discrete outcome given an input variable. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. We train following variants:

1. **Text Only**
   The model was trained using the TF-IDF representation of the tweet text only using combined Obama and Romney tweets. We find that character ngram-based model charLR performs better than word ngram-based model wordLR and use only character ngram representation in the next set of models.

2. **Meta Models**
   a. **meta_Combined** This model uses the tweet representation (char-based TF-IDF) and binary candidate information (Obama/Romney) as a meta feature, both fed to an LR classifier.
   b. **meta_Separate** Here we trained two separate LR models—one each for Romney and Obama using the tweet text (char-based TF-IDF) and used appropriate model at the time of validation. Since this information was expected to be provided for test set as well, we did not use further ensembling.

3.2 Transformer-based Language Models (LM)

A transformer is an encoder-decoder based deep learning architecture with self-attention[3] mechanism that has revolutionized the field of NLP. Large pretrained language models using transformer-based architecture have led to huge advances across NLP tasks (including text classification) spurred by the success of BERT[4]. BERT is trained on a large corpus of text in a self-supervised manner with masked-language-modeling and next-sentence-prediction objectives. Most importantly, unlike the bag-of-word models, it can capture the context that a term appears in.

---

[2] We also experimented with SVM and Naïve Bayes classifier on single split for initial experimentation with text-only models for similar results.

[3] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[4] Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *Proceedings of NAACL-HLT*. 2019.

Therefore, it will understand for example that the phrase “Man bites dog” has a different meaning from the phrase “Dog bites man” while both phrases will carry the same meaning using BoW models.

We find that a pretrained BERT model improves performance on one of the splits and motivated by this choose to apply RoBERTa[5] a state-of-the-art model on sentiment classification task. RoBERTa is built on BERT and optimizes its key hyperparameters, performs dynamic masking, removes the next-sentence prediction objective, and trains with much larger mini-batches and a lot more training data. We use both off-the-shelf RoBERTa model as well as fine-tune it for our data and find that fine-tuning brings large gains in performance.

4. Experiments

4.1 Implementation Details

BoW models: The ngram models were trained with ngram range (1, n). For word ngrams we experimented with n ranging from 1 to 6 and found that 5 gives the optimal performance, while for character ngram we experimented with range 1 to 12 and found 11 gives optimal performance. We tune the loss regularization term using manual search. The LR model was implemented using Scikit-learn library using cross-entropy loss. We also use balanced class weights to account for the skew in data.

Transformer-based LM: As an initial experiment we used off-the-shelf BERT using bert-base-case variant from huggingface.co. We also fine-tuned this model for 3 epochs. While it brought improvement over BoW models on one split. We proceeded with RoBERTa for full set of experiments as it had an even better performance on the same split and also since it is a larger model known to be state-of-the-art on twitter sentiment classification. We report scores with both—off-the-shelf variant cardiffnlp/twitter-roberta-base-sentiment from huggingface.co and its fine-tuned version over 2 training epochs.

Model	Accuracy	Macro-Average F1	Negative		Neutral		Positive	
			P	R	P	R	P	R
wordLR	58.52 (1.06)	57.86 (1.01)	65.15 (2.15)	62.93 (2.13)	52.76 (1.86)	50.6 (2.45)	54.98 (1.41)	61.24 (2.61)
charLR	59.05 (1.61)	58.3 (1.53)	65.13 (1.80)	64.17 (2.54)	53.89 (1.91)	51.34 (2.24)	55.34 (1.61)	60.26 (2.88)
meta_Combined	59.14 (1.52)	58.42 (1.42)	65.27 (1.85)	63.94 (2.39)	53.96 (1.90)	51.7 (2.51)	55.47 (1.5)	60.55 (2.79)
meta_Separate	59.31 (1.19)	55.29 (1.12)	64.72 (1.28)	64.87 (2.04)	53.39 (2.10)	**60.88 (2.29)**	57.35 (1.6)	60.88 (2.29)
RoBERTa off-the-shelf	56.53 (0.78)	53.49 (1.04)	68.03 (1.47)	65.34 (0.92)	44.67 (0.95)	64.55 (2.11)	63.56 (3.31)	30.33 (2.27)
RoBERTa fine-tuned	**71.12 (1.15)**	**70.27 (1.16)**	**73.49 (1.51)**	**80.13 (1.66)**	**66.6 (2.09)**	**59.19 (2.62)**	**72.06 (1.69)**	**71.09 (2.57)**

Table 2: Experimental Results averaged on 10-fold cross-validation with standard deviations reported in parenthesis. All the values are in %.

[5] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

4.2 Results

The performance of all the models is reported in Table 2. The scores are average across 10-fold cross validation. We see that using the binary candidate information in the meta_Combined model did not help much in improving the accuracy or the f1-score, although it is slightly better for the neutral class. The separate models are worse than single text-only models. However, this model has the best recall for neutral class. This indicates that the language use specific to sentiment expression is similar towards both the candidates. Off-the-shelf Roberta is better in terms of all metrics except precision for neutral class and recall for positive class, in which it is drastically worse. This makes it worst in terms of both accuracy and macro-F1 scores. Finally, we observe that RoBERTa fine-tuned has best performance across all metrics except in terms of average recall of the neutral class, where the separate logistic regression classifiers perform better. However, what RoBERTA fine-tuned loses in recall, it more than makes up in terms of precision making it the best model in terms of both accuracy and macro-average F1 scores. The F1 score is almost 12% more than the best BoW models. All the models perform better on Negative class as there is more training data available for this class. Surprisingly, this holds true even for RoBERTa off-the-shelf version which did not see the training data. We will have to investigate the distribution in original data it is trained on to see why this is so. Due to use of balance class weights, the gap between classes for BoW models is not as huge. The standard deviations are mostly around ±2 points, which indicates similar performance across the splits. We also conducted human analysis to see where the model makes error and could not determine anything concretely due to our limited domain knowledge of US politics.

We used the fine-tuned RobERTa from our third split during demo. This model has best F1 score (71.61%) of all. We consider F1 score for selecting our model due to the data imbalance, as in this case F1 becomes preferred metric over accuracy.

5. Conclusion

We find that fine-tuning a large language model known to have state-of-the-art performance on twitter sentiment classification gives the best performance on our dataset, more than 12% from the BoW models. The model variant without fine-tuning is worst. These experiments demonstrate the power of transfer learning and domain adaptation with fine-tuning. Overall, our models suffer from data imbalance. In the future we would like to experiment with under-sampling of negative class to alleviate this issue. We would also like to leverage tweets with mixed categories— creating duplicates with positive as well as negative labels and to see if this process improves the model performance. In addition, we would also like to experiment with word2vec as feature representation using methods that lie in between BoW and large LM in complexity. These include CNN and LSTM models that also consider the neighboring context, initialized with word2vec embeddings. Also, we did not leverage the timestamp information in current experiments as we could not hypothesize its connection to sentiment. However, we would like to test in future if people are more negative on some days such as weekdays or more positive at some times such as mornings or more ambivalent at some times when it comes to politics.