CS 583 Project 2 Report

Fangda Fan, Xiaohan Liu

Abstract

In this project, we cleaned the raw data collected from the tweets over the Obama vs. Romney election in 2012 and successfully classified them into three categories by political tendencies. Among the various non-sequential models we employed, LightGBM rendered the best overall accuracy of 65.69% on Obama involved tweets, and 62.83% accuracy on Romney involved tweets based on 10-folds validation. Moreover, in the sequential analysis, the Long Short-Term Memory and Convolutional Neural Network (LSTM + CNN) model scored a 65.46% accuracy on Obama's tweets and 63.61% accuracy on Romney's tweets. On the testing dataset, via LSTM + CNN, a 62.07% overall accuracy for Obama and a 65.47% overall accuracy for Romney was achieved. Finally, we constructed a linear mixed model to study each different dataset's, treatment's and ML model's contributions towards the overall accuracy.

1. Introduction

The dataset is 14,400 tweets related to 2012 US president candidates Obama and Romney (each about 7,200 tweets) during October 12-16, 2012. The sentiments of tweets toward candidates are labeled with -1 (negative), 0 (neutral), 1 (positive) and 2 (mixed). And we aim to do sentiment classification of tweets with label -1, 0, and 1 (2 is omitted).

After cleaning, there are 5625 tweets in Obama datasets, and 5648 tweets in Romney datasets. The distribution of labels is in the following table. The labels of Romney dataset are more unbalanced.

label	Obama	Romney
-1	1968	2893
0	1978	1680
1	1679	1075
Total	5625	5648

2. Technique

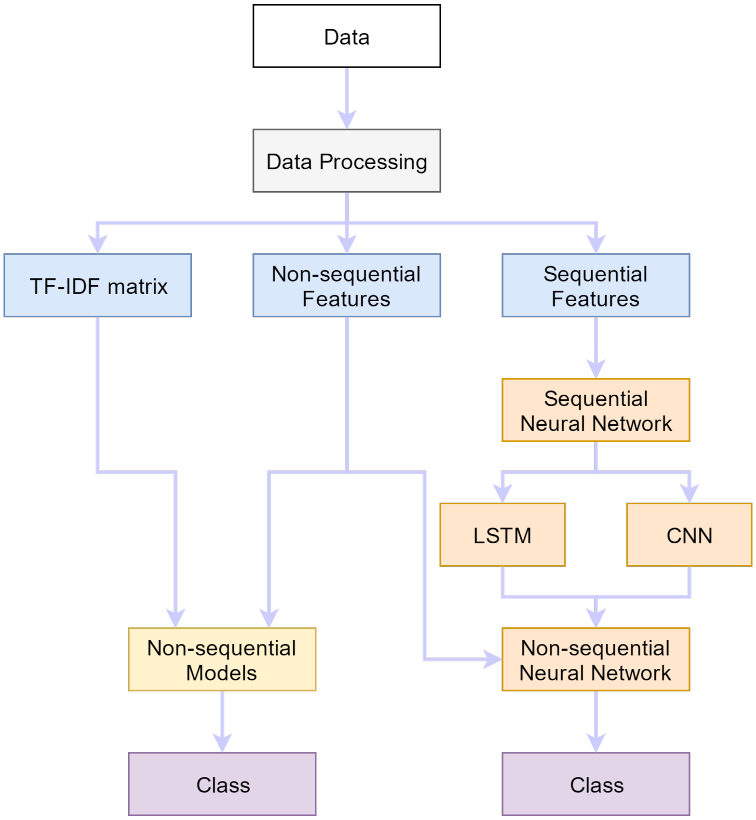
2.1 Data Processing

Firstly, we standardize date and time of the tweets into two numeric variables, because they probably would contribute to the classification and it did. Plus it is a good indicator of sampling quality and perhaps useful in future online learning when coupling with new data.

For annotated tweets, we extract English words, numbers, signs, punctuations, UTF-8 foreign words (including emoji), "<. . . >" HTML-tags, "@" mentions, "#" tags and URLs. Then for each extracted word, we do lemmatization, convert upper-case to lower-case, clean URLs to tag "<url>", and remove comma between digits. We use lemmatization instead of stemming with the hope of a lower level of artificial manipulation, also due to the sequential analysis procedures are not compatible with stemming. After cleaning, there are 10,852 unique words in Obama dataset, and 9,931 in Romney dataset.

After those pre-treatments, we create non-sequential and sequential features for each tweet. For non-sequential features, each tweet is represented as a vector. For sequential features, we analysis each word as a vector sequentially, and each tweet is 2-D tensor (word, position). The details are in section 2.2.

Finally, we randomly partitioned the data into 10 folds to do 10-fold cross-validation.



2.2 Features

2.2.1 Non-sequential features

We use TF-IDF to create the word matrix, each row is a tweet, and each column is a unique word. Most of the words have a very low support (frequency), so we remove all words with the frequency lower than 0.2% (about 10 tweets) when training model. There are around 800-900 variables left. Besides, we remove stopwords based on a 25 stopwords list.

We also do sentiment analysis, using Dr. Liu's opinion lexicon (Hu and Liu 2004) and VADAR sentiment analysis (Gilbert 2014). The use of opinion lexicon would add two more features to the previous matrix, one being the sum TF-IDF of all positive words, and the other being the sum TF-IDF of all negative. VADAR sentiment analysis is used to analysis the whole tweet and get positive, negative, neutral and compound scores. With also date and time, We add those 8 variables into the non-sequential tweet matrix for training.

2.2.2 Sequential features

We use GloVe (Pennington, Socher, and Manning 2014) pre-trained 2B tweets data as our word vector source. It represents each word as a vector of dimension 200 and can cover 63% of words in Obama dataset and 66% of words in Romney dataset. Also, SentiWordNet lexicon (Baccianella, Esuli, and Sebastiani, n.d.) help us get positive, negative, neutral scores of each word. Besides, we do part-of-speech (POS) tagging using Penn Treebank tags (Marcus, Marcinkiewicz, and Santorini 1993), creating a dummy variable of 20 columns for each word using the first two letter of Penn Treebank tags.

2.2.3 Treatment option comparison

In default, we do all data processing work and use all treatments in our models. To measure the effectiveness of each option on prediction accuracy, we make a comparison using different options with some treatments removed or changed. The following are changes compared with default:

- no datetime: remove date and time from variable list
- no sent: remove sentiment and opinion lexicon variables
- extract not: convert contraction about not (isn't, aren't, won't, ...) to not.
- no hashtag: remove '#' tag in front of words
- Non-sequential model only:
  - high freq: select word TF-IDF columns with word frequency greater than 0.5%
  - keep stopword: do not remove stopwords
- Sequential model only:
  - no POStag: remove POS tagging

2.3 Classification Models

We use non-sequential features for these models: Bernoulli naive bayes (BNB), linear SVM (SVM), AdaBoost (ADB), random forest (RF) and gradient boosting methods: gradient boosting trees (GB), XGBoost (XGB) (Chen and Guestrin 2016), LightGBM (LGB) and LightGBM DART (LGB DART) (Korlakai Vinayak and Gilad-Bachrach 2015). For random forest and AdaBoost, each model we use 500 trees; for 4 gradient boosting methods, each model we use 100 trees with learning rate 0.1 and max depth 5 (or max leaves 32), column ratio per tree 0.2-0.3.

A sequential model we choose is a combined neural network of LSTM and CNN (LSTM+CNN), it can use both sequential and non-sequential features. There are 4-6 hidden layers in the neural network (each layer with dropout), and the maximum nodes number in one layer is 256.

2.4 Contribution factors analysis

Consider the issue of analyzing contribution factors of accuracy for models and steps we have tried. Without missing prediction results, it need to run at least $n_{\text{dataset}} \times n_{\text{option}} \times n_{\text{model}} \times n_{\text{CV-fold}}$ to make a complete comparison, which is often a unnecessary way in practical. An alternative way is to make pairwise accuracy comparison using t-test along only one or two margins (model, option or dataset), but it only use a part of data collected every comparison and is hard to make a cross-margin comparison.

As a solution for that problem, we use linear mixed-effects model to make accuracy comparsion of different options and models on datasets across 10 CV folds:

$$Y_{hijk} = \beta_0 + X_{\text{dataset}(h)}\beta_{1h} + X_{\text{option}(i)}\beta_{2i} + X_{\text{model}(j)}\beta_{3j} + Z_{\text{dataset}(h),\text{fold}(k)}\nu_{hk} + \varepsilon_{hijk}$$

$$h = \{\text{obama, romney}\}, \quad i = 1, \ldots, n_{\text{option}}, \quad j = 1, \ldots, n_{\text{model}}, \quad k = 1, \ldots, 10$$

coefficient of fixed effects: datasets (β_{1h}), data cleaning options (β_{2i}) and models (β_{3j})

random effects: CV folds $\nu_{hk} \sim^{iid} N(0, \sigma_\nu^2)$ and residuals $\varepsilon_{hijk} \sim^{iid} N(0, \sigma_e^2)$

In the linear mixed-effects model, We assume each dataset, option or model has its overall fixed effects on accuracy (β's, need to be measure), and each CV fold in a dataset has its random effects on accuracy (ν_{hk}, needn't be measured), and each prediction has its independent own random error (ε_{hijk}, needn't be measured). The linear mixed-effects model needs no complete prediction results or complete CV folds to achieve its inference and is more powerful than pairwise tests.

3. Evaluation

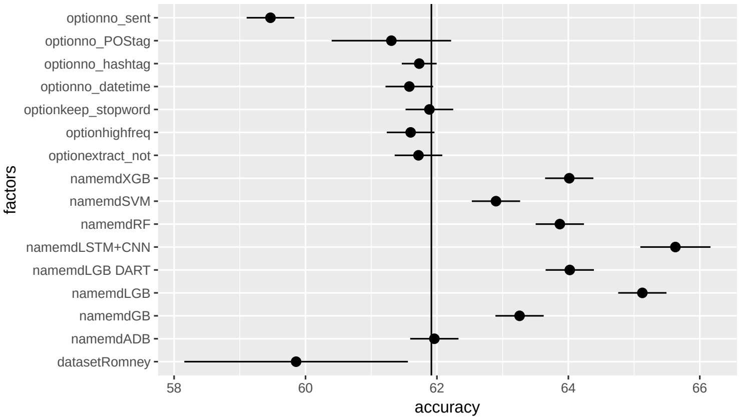
3.1 Training dataset with 10-folds CV

Table 2 is the estimation result of our linear mixed-effect model.

We use Obama as our reference dataset, complete data cleaning operation and as our reference option, and Bernoulli naive bayes as our reference model.

Table 2:

	Dependent variable:
	accuracy
datasetRomney	-2.06^{**} (-3.76, -0.36)
optionextract_not	-0.20 (-0.56, 0.17)
optionhighfreq	-0.32^{*} (-0.68, 0.05)
optionkeep_stopword	-0.03 (-0.39, 0.33)
optionno_datetime	-0.34^{*} (-0.70, 0.03)
optionno_hashtag	-0.19 (-0.45, 0.08)
optionno_POStag	-0.61 (-1.52, 0.30)
optionno_sent	-2.45^{***} (-2.81, -2.09)
namemdADB	0.05 (-0.32, 0.41)
namemdGB	1.34^{***} (0.97, 1.71)
namemdLGB	3.21^{***} (2.84, 3.58)
namemdLGB DART	2.11^{***} (1.74, 2.47)
namemdLSTM+CNN	3.71^{***} (3.18, 4.25)
namemdRF	1.95^{***} (1.59, 2.32)
namemdSVM	0.98^{***} (0.62, 1.35)
namemdXGB	2.10^{***} (1.73, 2.47)
Constant	61.92^{***} (60.68, 63.15)
random group SE	1.917
random error SE	1.258
Observations	760
Log Likelihood	$-1{,}299.33$
Note:	*p<0.1; **p<0.05; ***p<0.01



We see the choice of model has a great effect on overall accuracy, LSTM+CNN is the best model overall, has a 3.71% higher (95% confidence interval (3.18%, 4.25%)) accuracy than Bernoulli naive bayes, and LightGBM is the best unsequential models, has a 3.21% higher (95% CI (2.84%, 3.58%)) accuracy than Bernoulli naive bayes.

Besides, based on all the options we have tried, sentiment analysis has a great effect on overall accuracy, 2.45% (95% CI (2.09%, 2.81%)).

Based on 10-folds cross-validation, our best model LSTM+CNN has overall accuracy 65.46% in Obama and 63.61% in Romney, and these are the tables of precision, recall and F-score:

Table 3: Training 10-folds CV: Obama

label	precision	recall	Fscore	support
-1	0.64	0.73	0.68	1968
0	0.70	0.54	0.61	1978
1	0.64	0.70	0.67	1679

Table 4: Training 10-folds CV: Romney

label	precision	recall	Fscore	support
-1	0.69	0.80	0.74	2893
0	0.56	0.42	0.48	1680
1	0.57	0.54	0.56	1075

3.2 Testing dataset

When using the LSTM+CNN model on testing set. The overall prediction accuracy is 62.07% for Obama and 65.47% for Romney. And these are the tables of precision, recall and F-score of testing results:

Table 5: Testing: Obama

label	precision	recall	Fscore	support
-1	0.59	0.71	0.64	688
0	0.61	0.58	0.60	681
1	0.69	0.56	0.62	582

Table 6: Testing: Romney

label	precision	recall	Fscore	support
-1	0.71	0.78	0.75	960
0	0.53	0.50	0.51	555
1	0.66	0.57	0.61	385

4. Conclusion

We design a procedure to make sentiment classification on the tweets of 2012 president candidates, and evaluate performance of different datasets, treatments and models. We find that it is very useful to use outside data to help our prediction, such as sentiment lexicons and natural language models. Besides, sequential model based on LSTM and CNN shows a better prediction outcome than the models based on non-sequential bag-of-word.

Reference

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. n.d. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." In.

Chen, Tianqi, and Carlos Guestrin. 2016. "Xgboost: A Scalable Tree Boosting System." In *Proceedings of the 22Nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–94. ACM.

Gilbert, CJ Hutto Eric. 2014. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text." In.

Hu, Minqing, and Bing Liu. 2004. "Mining and Summarizing Customer Reviews." In *Proceedings of the Tenth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 168–77. ACM.

Korlakai Vinayak, Rashmi, and Ran Gilad-Bachrach. 2015. "DART: Dropouts Meet Multiple Additive Regression Trees." In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 489–97.

Marcus, Mitchell P, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. "Building a Large Annotated Corpus of English: The Penn Treebank." *Computational Linguistics* 19 (2). MIT Press: 313–30.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. "Glove: Global Vectors for Word Representation." In *EMNLP*, 14:1532–43.