SENTIMENT CLASSIFICATION

Obama-Romney Twitter Data

Submitted By-

Ahmed Metwally (UIN: 678120810)

Vishal Bansal (UIN: 669773290)

Abstract

Sentiment classification is one of the most fascinating and important area of research among modern text mining tasks. Twitter is a micro-blogging website with over 300 million users, who share their thoughts on this social sharing platform. If mined effectively, these opinions can help us to discover the untold story and unseen outcome. As the part of our CS583 – Data Mining and Text Mining, we applied some newest classification techniques on Obama-Romney Tweets data with the goal of classifying opinions into three main categories; Positive, Neutral and Negative. This report summarizes the methods that we applied to perform this task and our findings. The report contains a detailed description of preprocessing as well as methods of training various classification models, e.g., Naïve Bayes, Random Forest, and Deep Learning.

1. Introduction

Sentiment Classification is the task of finding the opinions and affinity of people towards the specific topic of interest. A similar task was assigned to us as the part of our project work. We were provided with the Obama-Romney-tweet dataset of labeled tweets from past election time of 2012. The goal is to leverage these labeled tweets to train a classification model which can be further utilized for predicting labels for any other data set.

1.1. Training Data

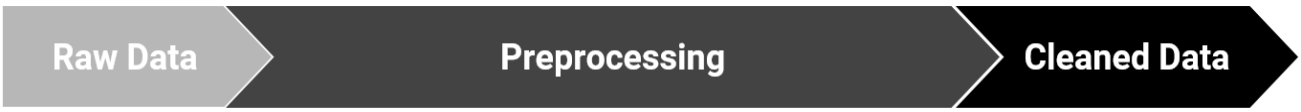
The data given to us has almost 7200 labeled tweets for both candidates. There are 4 labels present in the data; Positive (or 1), Negative (or -1), Neutral (or 0), and Mixed (or 2) along with some foreign labels such as 'Irrelevant,' 'Missing' etc. Image-1 shows the raw data. A lot must be cleaned before training a classification model.

	date	time	Anootated tweet	
			1: positive, -1: negative, 0: neutral, 2: mixed	Class
	10/16/12	09:38:08-05:0(	Insidious!<e>Mitt Romney</e>'s Bain Helped Philip Morri	-1
	10/16/12	10:22:34-05:0(	Senior <e>Romney</e> Advisor Claims <e>Obama</e> A(	2
	10/16/12	10:14:18-05:0(	.@WardBrenda @shortwave8669 @allanbourdius you m(	-1
	10/16/12	09:27:16-05:0(	<e>Mitt Romney</e> still doesn't <a>believe that we	-1
	10/16/12	10:11:43-05:0(	<e>Romney</e>'s <a>tax plan deserves a 2nd look b	-1
	10/16/12	10:13:17-05:0(	Hope <e>Romney</e> debate prepped w/ the same peop	1
	10/16/12	10:17:28-05:0(	Want to know how <e>Mitt Romney</e> is going to be al	-1
	10/16/12	09:35:55-05:0(	If <e>Romney</e> wins the <a>presidential election	-1
	10/16/12	09:33:07-05:0(	Presidential debate round 2: <e>Romney</e> wants a rep	2
	10/16/12	09:40:14-05:0(	Someone on the <e>mitt Romney</e> <a>Facebook page	!!!!
	10/16/12	10:28:50-05:0(	<e>Romney</e>'s <a>12 million jobs scam reminds r	-1

Figure 1: Raw Data

2. Preprocessing

Preprocessing of text data requires more effort compared to numerical data. Especially twitter data contains Usernames, URL's, Jargons, hashtags, etc. these elements need to be removed/treated as effective training model. The following pictorial description summarizes our preprocessing approach.



- ~7000 labeled tweets
- +ve, -ve, and Neutral labels.

- Elimination of Foreign, Mixed, and Irrelevant Labels.
- Removal of
  - Usernames, URL's - using **re**
  - Special Symbols, Punctuation Marks - using **re**
  - Foreign Characters ('ðÿ', 'œ', 'œi') - using **re**
  - Special Cases (e.g. rawww → raw) - using **re**
  - Stopwords (e.g. 'the', 'a', 'in') - using **nltk.corpus.stopwords()**
- Lemmatizing - using **nltk.stem.WordNetLemmatizer**
- Splitting - using **nltk.split()**

Class	Obama	Romney
+ve	30%	20%
-ve	35%	51%
Neutral	35%	29%
Total	5624	5648

It can be observed that Obama data has evenly distributed tweets of each label. However, Romney data has 51%. To balance the Romney data sampling methods must be used. Figure 2 shows data after cleaning.

['obama']	0
['go', 'ideas', 'hurt', 'obama', 'america', 'elect', 'blk', 'pres', 'youhave', 'show', 'whole', 'hand', 'dumbass']	0
['still', 'idol', 'mr', 'president', 'obama']	1
['pretty', 'creepy', 'wonder', 'obama', 'always', 'starbursts', 'pocket', 'equally', 'realistic', 'theories']	0
['saw', 'truck', 'nasa', 'sticker', 'obama', 'sticker', 'new', 'bff']	0
['debate', 'tonight', 'hope', 'obama', 'bring', 'lot', 'energy', 'tonight', 'debate', 'crucial']	0
['say', 'much', 'good', 'things', 'obama', 'small', 'business', 'owner', 'pick']	1
['hillary', 'take', 'spaken', 'obama', 'ignorance', 'attaboy', 'obama', 'blind', 'people', 'back', 'better', 'spineless']	-1
['south', 'african', 'say', 'obama', 'best', 'thing', 'happen', 'america']	1
['smart', 'women', 'know', 'obama', 'say', 'policies', 'would', 'cause', 'energy', 'price', 'skyrocket', 'mean']	-1
['tonight', 'obama', 'focus', 'forceful', 'leave', 'voters', 'wonder', 'not', 'know', 'first', 'debate']	0
['obama', 'may', 'need', 'grind', 'game', 'backers', 'follow', 'elex', 'closely', 'compare']	0

Figure 2: Cleaned Data

3. Training Model

We trained 3 classification models using Naïve Bayes Classifier, Random Forest Boosting, and LSTM. We have one model representing each candidate. We are going to discuss the LSTM model in detail here.

3.1. Long Short-Term Memory

LSTM is capable of learning long-term dependencies. LSTM is very successful in speech recognition, language modeling, and translation. The key to LSTM is the cell state. Figure 4 shows the architecture we used in our LSTM model. We used Soft-max Cross Entropy as the loss function. The network is trained using Adam optimizer, regularized using Dropout [1], and implemented in TensorFlow. Each tweet word is represented in 50-dimensional representation. We used the GloVe database [2] to map each word to the corresponding vector representation. For the hyperparameters, we used Batch Size = 25, LSTM Units: 50. We ran the network for 50,000 iterations, and we save the model every 500 iterations. The best model is restored to do the prediction. Figure 4 shows how the model behaves on training and testing datasets.

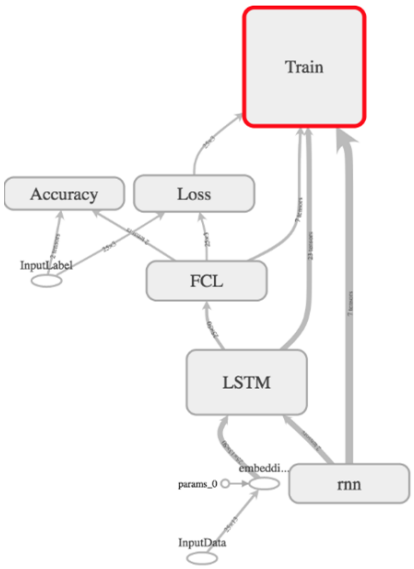


Figure 3: The used LSTM architecture

Figure 4: LSTM Training/Testing Accuracy and Loss. Pink and blue lines represents training and testing datasets, respectively.

4. Evaluation

Our model's evaluation is done based on 10-fold cross-validation. Figure 5 and 6 summarizes the result obtained from the 3 classifiers.

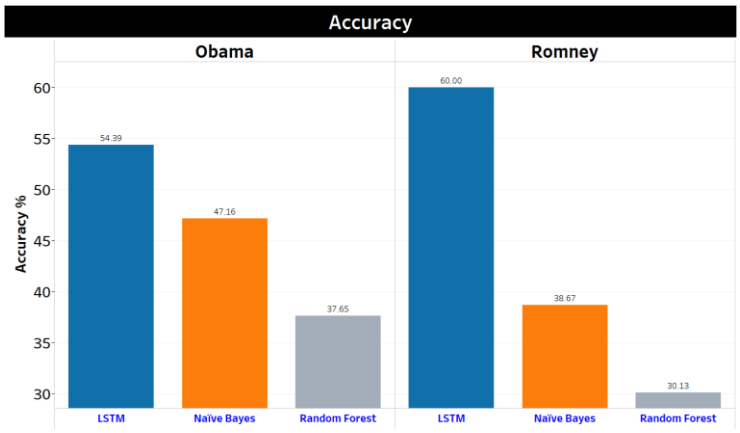


Figure 5: Evaluation of accuracy for LSTM, NB, and RF

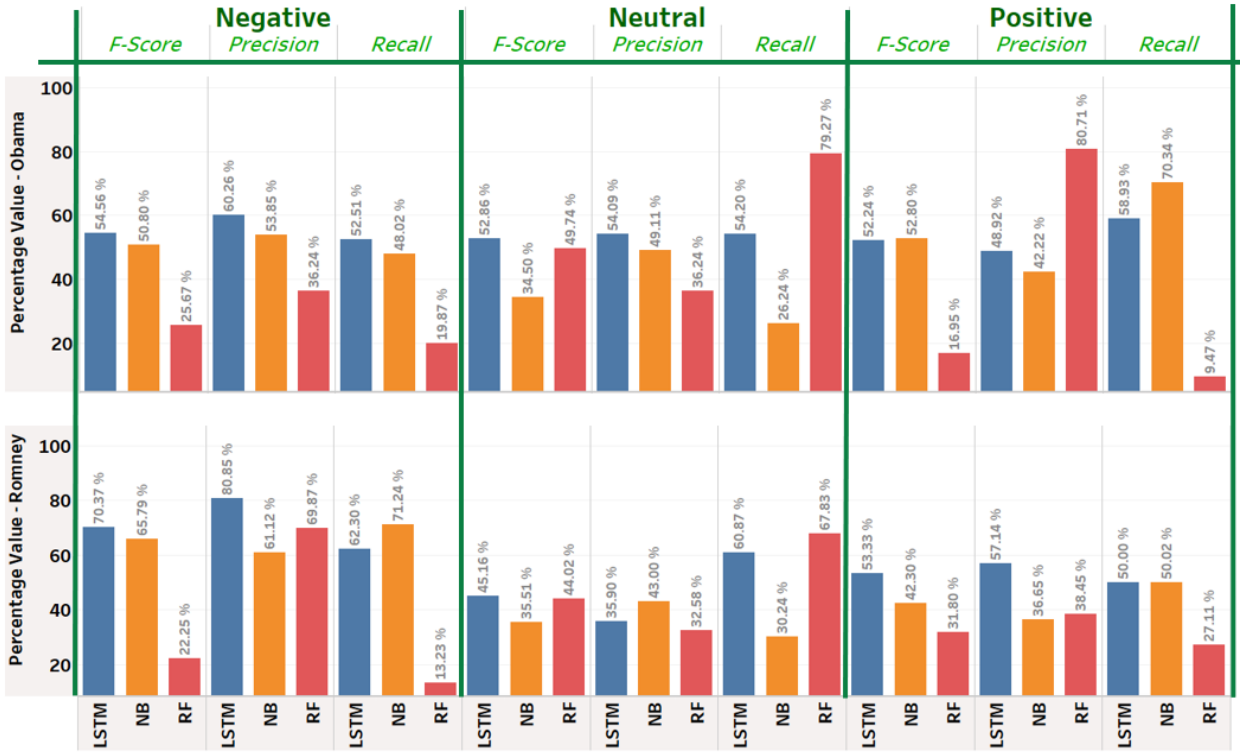


Figure 6: Evaluation of F-score, Precision, and Recall for LSTM, NB, and RF

5. Conclusion

LSTM provided us with a very high evaluation metrics. The reason for the decrease performance on the Romney positive data is because the datasets are skewed as shown in the preprocessing step, and we didn't balance the data in the training and testing procedure. Also, the reason behind the poor performance of the NBC and RF is that we didn't spend much time optimizing their parameters since our focus from the beginning was toward building the LSTM model.

6. References:

[1] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of machine learning research 15.1 (2014): 1929-1958.

[2] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. APA