An Evaluation of Tweet Sentiment Classification Methods

Lihua Yao[1]
Department of Defense, OPA, USA
Lihua.Yao.civ@mail.mil
Jerry Li
DeepHole AI Consulting, USA,
jerryirc99@gmail.com

Hassan Alam
AI Asset Management, USA,
hassan_alam2003@yahoo.com
Oleg Melnokov
Stanford University, USA
oleg2@stanford.edu

Abstract

In this paper, we present the result of our research in predicting sentiment from Twitter data derived from a Kaggle competition. Our goal was to determine the efficacy of different supervised classification methods to predict Twitter sentiment to be Positive, Neutral or Negative. We evaluated four different classification statistical models: 1. Logistic Regression (LR), 2. Linear Support Vector Machine (LSV), 3. Multinomial Naïve Bayesian (NB), and 4. Random Forest (RF). We also evaluated two different tokenization methods 1. Document Term Matrix (DTM) and 2. Term Frequency–Inverse Document Frequency (TF-IDF). We combined this with three text extraction methods 1. Original Tweet Text, 2. Rapid Automatic Keyword Extraction (RAKE) and 3. Hand curated "Selected Text". Furthermore, various Neural Networks were applied to the Tweet Text and BERT extracted data that reduced the original 1000 features to be 768 that were applied to different models. Our experiment shows RF and LR gives the best results and there is little difference between DTM and TF-IDF. Fully connected neural network (FCNN) performed the best for the BERT extracted data with a test score of 0.75.

Keywords

Processing, NLP, TF-IDF, Statistical Models, Neural Network, Twitter.

Sentiment Analysis

Sentiment Analysis (also known as opinion mining or emotion AI) uses Natural Language Processing (NLP) to automatically identify the emotions, opinions, or attitude from text corpus. The process generally involves four steps:

1. Cleaning the data to standardize it and remove unnecessary information.
2. Pre-processing the text corpus: terms within the corpus are normalized which might contain the following steps: a) Convert to lowercase or upper case. b) Stemming and lemmatization. c) Stopword removal.
3. Feature extractions and tokenization.

[1] Contact Author Address: Lihua Yao, Department of Defense, Office of People Analytics, Assessment Center, 400 Gigling Rd, Seaside, CA,993955; Lihua.Yao.civ@mail.mil

The views expressed in this paper are those of the authors and not necessarily those of the Department of Defense or the United States Government.

978-1-7281-7624-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCI51800.2020.00057

The unstructured text is transformed into a structured form. The most commonly used method is "Document Term Matrix," (DTM), in which documents are treated as rows (i.e., observations) and the items of text are treated as columns (i.e., features or terms). The element *TF(ij)* for the ith row, *jth* column in this matrix is the frequency of appearance of the *jth* term in the *ith* document. Another popular method is Term Frequency-Inverse Document Frequency values (TF-IDF) (Silge & Robinson, 2018). TF-IDF generates values by first calculating a Term Frequency (TF) for the number of times each term occurs within a document. Next, each TF value is multiplied by an Inverse Document Frequency (IDF) value that assigns higher weight for elements that are rarely found across documents. The Document Frequency is computed by the number of documents that the term appears in divided by the total number of documents. Therefore, if a term is infrequent, and only appears, for example in one document, then IDF will be higher.

4. The fourth step of the sentiment analysis process is deciding upon how to classify each document's overall sentiment. If the data sets have previously labeled sentiment, then test errors for the sentiment based on the trained predictor from the training data can be used as a measurement for the sentiment analysis. Statistical models are applied in this step.

There are many methods for sentiment analysis in tokenization and feature extraction; for example, Bag Of Words BOW, TF-IDF, and word embeddings (Das & Balabantaray, 2014; Le & Mikolov, 2014), parts-of-speech (POS, Kharde & Sonawane, 2016), BERT (Google), and HAN (Carnegie Mellon University). There are many statistical models fitting the data and predicting the sentiment. We used 4 models to see which one would give better results:

Logistic Regression (LR):
regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

Linear Support Vector Machine (LSV):
In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems.

Multinomial Naïve Bayesian (NB):
Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models.

Random Forest (RF):
random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Besides the four statistical models, natural network such as Fully connected neural networks (FCNNs), Fully connected neural network with

embedding, Convolutional neural network, Recurrent Neural Network, and Convolutional+Recurrent Neural Network were applied to the original tweet text data and BERT extracted data.

Data Sets

For the purpose of our experiment we obtained our data sets from Kaggle that can be found at *https://www.kaggle.com/c/tweet-sentiment-extraction/data*.

The data is divided into training data and test data.

1. Training set (train.csv 3.34 MB). 2. Test Set (test.csv 307 kb),

The data has four columns:

- "textID" – unique alphanumeric ID for each Tweet. For example, for the first observation, textID is "549e992a42",
- "text" - the text of the tweet. There may be few words, with a maximum size of 240 Chars. For example, "Sooo SAD I will miss you here in San Diego!!!"
- "sentiment" - the general sentiment of the tweet that has three categories: "positive", "neutral", and "negative".
- "selected_text" - [training set only] the text that supports the tweet's sentiment. Example for the tweet with "text": "my boss is bullying me...", the "selected_text" is "Sooo SAD". This is manually extracted sentiment text from the tweet itself.

The sample size for the training data and the test data are (11117, 8582, 7781) and (1430, 1103, 1001), respectively, for the three categories "neutral", positive", and "negative". To determine if aggregation of sentiment would give higher accuracy, we created another column 'label ` that combined "positive" `` and "negative", and classified the tweets into two categories. This column identified if a sentiment exists (positive or negative) or doesn't exist (neutral).

Data Handling and Preparation

Data Cleaning

Like all tweets the data has the following characteristics: Incomplete sentences and Twitter specific jargon (emoticons, slang, word shortening etc.). In this experiment we did basic cleaning such as removing trailing and leading space; delete tweets with NAN or with only one character.

Feature Extractions and Tokenization

For our experiment, each of the individual tweets "textID" in the training data is the rows or documents. The term or the features or the columns were extracted from columns of the training data listed below: 1) Original tweets, the column "text" in the data set. 2) Rake-extraction technique extracted from "text", named "Rake_Text". 3) "selected_text".

"Rake-Extracted" column contains words extracted from "text". RAKE is short for Rapid Automatic Keyword Extraction algorithm, is a domain independent keyword extraction algorithm, which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrence with other words in the text.

To show what the data looks like, two observations with columns (textID, "text", "selected_text", "Rake_Extracted","sentiment", "label")

are: (“549e992a42”, “Sooo SAD I will miss you here in San Diego!!!”, “Sooo SAD”, “miss, san diego !!, sooo sad”, “negative”,”1”) and (“cb77db0d1”,” I`d have responded, if I were going”,” I`d have responded, if I were going”, “going, responded”,”neutral”,”0”), respectively.

For each of the feature variations, DTM and TF-IDF processes were applied; the train data applied to models had 27480 rows and 1000 columns.

Sentiment for supervised classification

1. “Sentiment” that has three classification categories (Positive, Neutral, and Negative) is used in the training process for all emotions.
2. “Label;” that has two classification categories is used in the training process for the absence or presence of emotion.

Results and Summary

In our research we used DTM and TF-IDF for tokenization; LR, LSV, NB, and RF statistical models were applied in fitting the training data. For evaluating the results of each condition, we used classification accuracy, precision, and recall measures on the test data. Test score or classification accuracy for each condition is listed in Table 1 below.

Table 1: Accuracy for the training and test data with three sentiment levels

Token	Model	txt_train score	txt_test score
DTM	LRG	0.719	0.697
DTM	LVC	0.715	0.692
DTM	NB	0.684	0.669
DTM	RF	0.968	0.675
TF-IDF	LRG	0.712	0.697
TF-IDF	LVC	0.711	0.691
TF-IDF	NB	0.667	0.654
TF-IDF	RF	0.967	0.693
Token	Model	selected_ train score	selected_ test score
DTM	LRG	0.802	0.593
DTM	LVC	0.795	0.59
DTM	NB	0.742	0.629
DTM	RF	0.908	0.597
TF-IDF	LRG	0.766	0.591
TF-IDF	LVC	0.8	0.571
TF-IDF	NB	0.751	0.604
TF-IDF	RF	0.907	0.523
Token	Model	Rake_trai n score	Rake_tes t score
DTM	LRG	0.719	0.694
DTM	LVC	0.714	0.687
DTM	NB	0.684	0.66
DTM	RF	0.968	0.676
TF-IDF	LRG	0.703	0.686
TF-IDF	LVC	0.707	0.688
TF-IDF	NB	0.66	0.655

We combined positive and negative sentiment to give two classes of sentiments – existence of sentiment (pos-neg) and no sentiment (neutral); results are listed in Table 2. It can be seen that test scores for the two sentiment levels are higher than test scores with three sentiment levels. We observed that RF has the best training data score and test data score when using “text’ for sentiment prediction, followed by LR. RF emphasizes feature selection – i.e. weighs certain features as more important than others. It does not assume

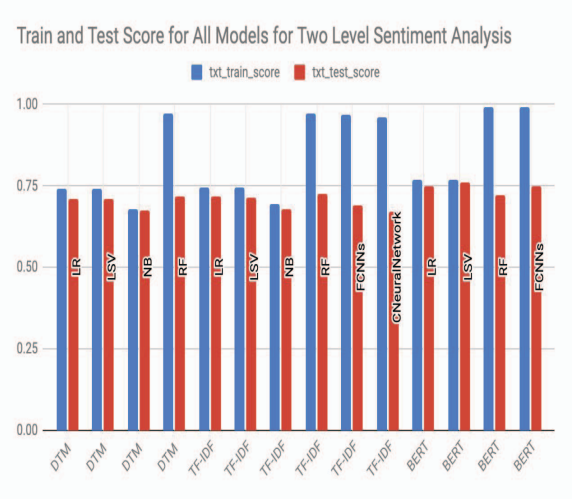
a linear relationship, unlike the Regression Models. A random forest takes random samples, forms multiple decision trees instead of just using one, and then averages out the leaf nodes to get a clear model. We are taking on averages of 200 tree samples for our experiment. Higher training score for RF is not uncommon. High training score and low test score may be because of overfitting or maybe because of differences between training data and test data.

Table 2: Accuracy for the training and test data with two levels sentiment

		txt		selected_text	
		train score	test score	train score	test score
DTM	LR	0.742	0.711	0.836	0.575
	LSV	0.739	0.709	0.837	0.571
	NB	0.68	0.673	0.831	0.631
	RF	0.97	0.716	0.952	0.58
TF-IDF	LR	0.744	0.717	0.687	0.627
	LSV	0.744	0.715	0.842	0.58
	NB	0.694	0.679	0.795	0.662
	RF	0.97	0.725	0.953	0.518

Besides the four statistical models mentioned previously, fully connected neural networks (FCNNs) were applied to our data with two classifications. With epochs=30, and batch_size of 128, the training score and the test score were (0.967, 0.69) and (0.946, 0.571) for "text" and "selected-text", respectively. FCNNs did not perform better than the four statistical models. For "text", both FCNN with embedding dimension of 50, and Convolutional Neural Network had training scores and test scores of (0.96, 0.67). However, the test score for Recurrent Neural Network was only 0.59.

BERT extraction technique was applied to the "text" data: RF was applied and the train score and test score were (0.99, 0.72), respectively. LR and LSV had test scores of 0.75, 0.76, which were much higher than the others, but their train scores were only about 0.77. FCNN was applied to BERT extraction vectors, the train score and test score were (0.99, 0.75), respectively. See the Figure below that summarized two level results for all models.



We computed Precision and Recall and F score for TF-IDF. Table 3 lists the F score for the four models that were optimized for their parameters for each type, for the three sentiments. We observed that "positive" had the highest F scores and "negative" had the smallest F scores for all four models. On average, RF had the highest F score.

Table 3: F scores for the four models

	negative	positive	neutral
LR	0.6332	0.7381	0.6822
LVC	0.6375	0.7455	0.6832
NB	0.5984	0.6831	0.6655

RF	0.6504	0.7415	0.6783

Discussion of Methods and Models

In this tweets data analysis, TF-IDF did not yield better results than DTM, as other research demonstrated. TF-IDF has an advantage over DTM in situations where some features appear infrequently in documents. Maybe the sample size for the training is not large enough, so there are not enough infrequent terms. From the results we observe that two level classifications yielded better scores than three level classification. We hypothesize that larger training size would improve the test scores for three level classification predictions and also would improve results for TF-IDF. For tweets data, RF consistently performs better than other models; other variations of RF might be applied to decrease overfitting. FCNNs with BERT extraction had the highest train score of 0.99 and test score of 0.75, among all the variations.

We would like to investigate the effect of Part-Of-Speech (PoS) tagging on Twitter Sentiment Classification. POS tagging is the process of marking words in text with their Parts-Of-Speech (adjectives, adverbs, nouns, etc.) to give a better understanding of their meanings within the context that they appear. For example, marking verbs and adjectives may help focus a sentiment analysis model on important terms. Verbs and adjectives commonly have greater value for predicting sentiment than other parts of speech like nouns and prepositions.

We also would like to determine the effect of introducing a "Tweet-Sentiment" Dictionary (Farooqui etc, 2019). In our research, we conducted supervised classification using the sentiment. The training data set has a column titled "selected_text" which is a subset of the original tweets. We want to research how we can train the machine to use that data to produce the "selected_text" (i.e. sentiment phrases) for the test or future tweets. We would apply Word Embedding Layers for Deep Learning with our Pre-Trained model with the dictionary.

References

Das, O., & Balabantaray, R. C. (2014). Sentiment analysis of movie reviews using POS tags and term frequencies. *International Journal of Computer Applications, 96*(25).

Kharde, V., and Sonawane, S. (2016). Sentiment analysis of twitter data: A survey of techniques. *International Journal of Computer Applications, 139*(11), 5-15. doi:10.5120/ijca2016908625

Le, Q., and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196).

Nafees A. Farooqui, Ritika, & Aayush Saini (2019). Sentiment Analysis of Twitter Accounts Using Natural Language *Processing International Journal of Engineering and Advanced Technology (IJEAT) 8(3),pp.* 2249 – 8958

Silge, J., and Robinson, D. (2018). Analyzing word and document frequency: TF-IDF. In *Text Mining with R: A Tidy Approach*. Retrieved from https://www.tidytextmining.com/tfidf.htm