Sentiment Analysis of Twitter Message using Doc2vec

Joshua Acosta and Yangyang Chen

**Abstract**—This study tries to implement a sentiment analysis on a group of Twitter posts concerning the first Republican Party Presidential debate in the year of 2016. The goal is to classify users’ attitudes into three categories by using Doc2vec, an extenstion of shallow neural network models called Word2vec. The methodology tries to create vectors from a group of words and measure their distance between each other. The experiment shows that the highest accuracy for this analysis reached 62% by using the Synthetic Minority Oversampling Technique (SMOTE SVM) as the sampling method and distributed memory (DM) as the training method. There are still challenges remained for future studies in terms of increasing the accuracy rate.

**Index Terms**— Doc2vec, multiclass classification, natural language processing, sampling technique, sentiment analysis, Twitter and word embedding.

—————————— ◆ ——————————

# 1 Introduction

A

S one of the biggest social network services, Twitter proved itself to be an important source for people expressing their opinions on current happenings and hot issues, one of which is indeed on this year’s election. On the day of the election, Twitter is said to be “the largest source of breaking news, with 40 million tweets sent by 10 p.m. (East Coast Time)” [5].

It has already known that sentiment analysis—analyzing one’s attitude through his/her’s wording—can be significant in helping decision makers to better evaluate what public’s inclination is and possibly have an impact on their critical decisions. Therefore this platform provides significant datasets for measuring what people think about the president election of this year. This paper selects tweets that talk about first Republican Party Presidential debate as the primary dataset. We try to use doc2vec, an extention of word2vec models, to implement a sentimental analysis on this group of data.

As part of the Natural Language Process field, studies on sentiment analysis have been around for decades, with different methodoloties emerging from time to time. More traditional ways of determining sentiment includes Max Entropy, Support Vector Machines, Naïve Bayes, etc. A very common way of measurement is to determine to which polarity a message is closer: positive, negative, or neutral. This is also the criteria of classification for this paper. Other methodologies includes determining subjectivity/objectivity, feature/aspect identification, just name a few.

The method we chose for this sentiment analysis is doc2vec, an extention of word2vec created by a team led by Tomas Mikolov at Google. Word2vec is not a single algorithm, but rather a group of models belonging to the shallow neural networks. It produces a vector space with its inputs. Each word is assigned a vector in the space and positioned. Words that share common contexts are closer to each other in this space. Instead of targeting individual words, doc2vec creates embeddings from a group of words. The languages applied to this methodology includes C, Java, and Python, the latter is what we use for this research.

Numerous researches have been done regarding the use of word2vec while there are only a small number addressed doc2vec. Our attempt is to further explore this methodology and obtain an in-depth understanding on its application.

# 2 Literature Review

Though doc2vec is not so widely mentioned, its predecessor word2vec has been implemented and explored substantially. Many of the researches have set the subject on sentiment expressed through social media. These provide sufficient reference for our study.

In a research led by Lilleberg, word2vec is combined with tf-idf, or term frequency-inverse document frequency, to add the influence of word’s frequency beside of the space between vectors when determine the sentiment expressed in a text. The higher tf-idf values are, the stronger relationship words have in the document [1]. Different combinations of techniques of using word2vec and tf-idf have been experimented. It turns out that when weigning word2vec with tf-idf without stop words and use it to work with tf-idf withough stop words, the accuracy appears to be the highest. Though this high score of accuracy is not guaranteed everytime, the average performance of this approach is still favorable. The research shows that word2vec can actually offers a lot when it comes to text classification, especially when incorporating it with other techniques [1].

A research that can almost correspond to ours is a sentiment analysis of Sina Weibo with word2vec conducted by Bai Xue, Chen Fu, and Zhan Shaobin. Sina Weibo, or micrio-blogging is a social networking service widely used in China. It can be regarded as the equivalent of Twitter. The classification is also polarization based. They first established a sentiment dictionary based on Chinese words’ Sentiment Orientation (SO) [6]. One thing to note is that they also took cyberwords into consideration while building the dictionary. Then an algorithm is created with an idea similar to K-Nearest Neighbor (KNN) to determine words’ semantic orientation. Similarity distance between two words are calculated to see if a word is more positive or negative oriented. Also, the intensity of the sentiment is calculated. It shows that “sentiment with high sentiment value” can help with improving the accuracy rates [6]. The authors point out an important fact about the nature of (natural) language. That is, words and their usage are constantly changing. In an internet environment, this flexible and sometimes improvisional change of a word’s meaning and way of use is a main reason causing the sentiment dictironay limited. It might be necessary to construct a specific dictionary based on a particular social media platform for certain words can only be used in a certain way there. Moreover, it is important to notice that a dictionary “should not be definite but should be extended and imporved all the time” [6]. Like the previous article, this research also combines word2vec with another methodology called Semantic Orientation Pointwise Similarity Distance (SO-SD) to obtain a better predition results.

Duyu Tang, et.al reviewed different methodologies including word2vec regarding the sentiment analysis in their research of sentiment embeddings with application. They studied sentiment-specific word embeddings in the aspect of the ability of word embeddings in capturing word similarities interms of sentiment sematics. The result shows that “words with similar context but opposite sentiment polarity lables such as “good” and “bad” can be separated in the sentiment embedding space”[3]. Several neural networks were introduced to encode the context and sentiment level information into word embeddings in a “unified way”. Their research showed that sentiment embeddings are: useful for discovering similarities between sentiment words on a word leve; helpful in capturing discriminative features for predicting the sentiment of sentences on sentence level; helpful in measuring the similarities between words on a lexical level. Besides word2vec, C&W, SE-Pred, SE-rank, and other models are examined and compared. Interestingly, the conclusion also indicates that a “hybrid models that captures both context and sentiment information are the best performers…” [3]

In a reaserch about Twitter Sentiment Classification using Distant Supervision, Alec Go et.al mentioned the use of Unigrams, Bigrams and Parts of Speech as the feature extractor when dealing with the highly flexibility of languges. [4]

Havin reviewed a number of related works on sentiment analysis, espcecially with word2vec, it can be seen that this approach is often, and preferably used while either combining with othe techiniques or methodologies, or having part of the methods modified to reach for a better results. Considering the characteristics of internet world and the fact of fast-emerging memes everyday, language used on social media like Twitter should be carefully examined based both their context and sentiment as an attempt to minimize the ambiguity.

Given the fact that doc2vec is an extention of word2vec, we assume the conclusions from word2vec studies should have a similar application or at least important guidance towards our research.

# 3 Methodolody

In order to study the sentiment of Twitter data, the experiments were conducted on a publicly available dataset ([www.kaggle.com](http://www.kaggle.com)) of Twitter posts relating to users’ experiences while watching the first Republican party Presidential debate. The dataset contained 13,871 tweets and 21 attributes including the original tweet text, Twitter user-related data, and the class sentiment label. Before training machine-learning models on the data, some exploratory data analysis was conducted on the dataset to get a better understanding of what it entailed. Our exploratory analysis showed an imbalance between the three sentiment classes with 8493 labeled as negative, 3142 labeled as neutral and 2263 labeled as positive instances. The imbalance of classes was dealt with before classification and further discussed in this and later sections.

According to the Kaggle site, for this dataset, the users were asked to describe their tweet as positive, negative or neutral and whether it was related to the debate. These classes represent the criteria by which the accuracy of the experiments were analyzed. The analysis also showed a higher number of negative instances for each of the 10 candidates (Donald Trump, Ted Cruz, Jeb Bush, Marco Rubio, Rand Paul, John Kasich, Chris Christie, Miek Huckabee, Ben Carson and Scott Walker) than positive and neutral which can also cause for instances to lean towards being classified negative. For Donald Trump, Mike Huckabee, Jeb Bush and Chris Christie, there are much more negatives tweets for each candidate than there are positive or neutral tweets which can affect the training for how the classifiers determine the sentiments of the test data or future posts. Candidates like Ted Cruz, Marco Rubio and John Kasich has more positive reviews than negative or neutral but won’t affect the training of classifiers due to the overwhelming number of negative response overall.

The scikit-learn’s *train\_test\_split* function was used to split the Twitter posts and their respective sentiment labels. 70% of the samples were allocated for training and 30% for testing purposes.

Before training the doc2vec model, the data was cleaned up with the use of a function that returns a list of sentences, where each sentence is a list of words. For each sentence, all the uppercase letters and words were changed to lower case letters to ensure uniformity. In addition, all the words in a sentence were changed into individual strings, and all of the extra spaces around the words were truncated. In order to produce the word embeddings using doc2vec, the tweets are required to be labeled before training. The tweets were provided either a “TRAIN” or “TEST” label to satisfy the requirement. As recommended by the creators the algorithm, the

In order to compare the results of the different trials, two different training algorithms were instantiated: 1) one model using distributed bag-of-words (DBOW) and 2) another model using distributed memory (DM). The training algorithms were instantiated using the sentences of tokenized words from each tweet and configuring parameters. These parameters included the number of vector dimensionality features set at 300, the context window size around words set at 10, and the minimum word count of words that was set to five before the samples were included in the model. The models were tested using both *hierarchical softmax* and *negative sampling*. Hierarchical softmax provided higher accuracy scores and therefore it was chosen.

Before using the classifiers to predict sentiment the imbalances within our classes were resolved. Several statistical methods can be implemented to address this challenge such as oversampling, undersampling, synthetic minority oversampling and adaptive synthetic sampling. Since our dataset size was limited, several sampling techniques were used to select random instances from the underrepresented classes and duplicate them until there was an equal amount of instances in the training set for each class. This technique allowed us to avoid instances being classified as the majority class often enough to skew results.

Before training the classifiers, the vectors were then scaled using scikit-learn’s scale function from its preprocessing library. The scaled vectors were then used to train a Support Vector Machine classifier, results were recorded and are discussed in the results section.

# 4 Results

# 4.1 Initial Results

Similar to the survey conducted by Lilleberg [1], our study compares the accuracy of Support Vector Machines machine learning algorithm. As shown in Table I, the classifier was trained with two different training models, DBOW and DM, embedded inside the doc2vec algorithm.

TABLE I. Accuracy Percentage by Classifier and Training Model

|  |  |  |
| --- | --- | --- |
| Classifier | Training Method | Accuracy % |
| Support Vector Machine | DBOW | 51 |
| Support Vector Machine | DM | 51 |

As shown in Table I, both training models trained with Support Vector Machine classifier using a linear kernel produced an accuracy score of 51%. Upon further review, the similar score was caused by an imbalance of negative training tweets therefore classifying most test tweets as negative.

In order to obtain meaningful results, we compared several sampling techniques to combat the imbalance in the dataset. Xu et al. [2] introduced the Synthetic Minority Oversampling Technique (SMOTE) creating synthetic samples from the minor classes instead of creating copies similar to simple oversampling would do.

# 4.2 Sampling Results

TABLE I. Accuracy Percentage by Sampling Method and Training Model

|  |  |  |
| --- | --- | --- |
| Sampling Method | Training Method | Accuracy % |
| Undersampling | DBOW | 46 |
| Undersampling | DM | 47 |
| Oversampling | DBOW | 51 |
| Oversampling | DM | 51 |
| ADASYN | DBOW | 52 |
| ADASYN | DM | 54 |
| SMOTE SVM | DBOW | 62 |
| SMOTE SVM | DM | 62 |

As shown in Table II, all the oversampling methods underperformed the initial results with the imbalanced classes except the SMOTE SVM. The DBOW and DM models which were undersampled produced an accuracy score of 46 and 47 percent respectively while simple oversampling produced a score of 51 percent. The best performing method was SMOTE SVM producing a small increase over the initial result at 62 percent.

# 5 Conclusions

Social media applications like Twitter and Facebook have given users the ability to share their opinions to billinos of users around the globe. It has also provided companies with enourmous amounts of feedback regarding satisfaction with product, policies and/or procedures. With millions of users going on Twitter to express their opinions, Twitter is an ideal platform for sentiment analysis.

Sentiment analysis the process of classifying whether a body of text conveys a positive, negative, or neutral sentiment. One of the challenges about sentiment analysis on Tweeter posts is its 140-character limit per post. After subtracting the length of a user’s twitter handle, there isn’t much space left to complete a proper sentence. Typically, a user abbreviates words whenever possible, uses slang or uses a hashtag, which doesn’t allow much space for content.

This study attempted to determine whether using the doc2vec algorithm to create word embeddings could be used to classify sentiment. By using the word embeddings, our researchers would avoid having to manually create features based off stylometry, parts-of-speech tagging, create dictionaries, etc. in order to classify correctly. The dataset was acquired from Kaggle.com, which contains over 13,000 tweets about the first Republican Party Presidential debate and their classification. Due to the imbalance among the sentiment classes, our dataset used several sampling techniques to improve the imbalance.

Support Vector Machine classifier was used in an attempt to classify over 3,000 tweets after training the doc2vec algorithm with over 9,000 tweets. The highest accuracy rate yielded by a classifier was 62% using Support Vector Classifier and both DM and DBOW training model.

**References**

1. Lilleberg, Joseph, and Yun Zhu. "Support Vector Machines and Word2vec for Text Classification with Semantic Features." *Cognitive Informatics & Cognitive Computing* (2015): n. pag. Print.
2. R. Xu, T. Chen, Y. Xia, and Q. Lu, "*Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification,*" 2015. [Online]. Available: https://www.researchgate.net/publication/271856480\_Word\_Embedding\_Composition\_for\_Data\_Imbalances\_in\_Sentiment\_and\_Emotion\_Classification. Accessed: Dec. 19, 2016.
3. Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, Ming Zhou, "Sentiment Embeddings with Applications to Sentiment Analysis", *IEEE Transactions on Knowledge & Data Engineering*, vol.28,no.,pp.496-509, Feb. 2016.
4. A. Go, R. Bhayani, and L. Huang. “Twitter Sentiment Classification using Distant Supervision”. [Online]. Available: <http://www.citeulike.org/user/pcalado/article/8047905>.Accessed: Dec. 12, 2016.
5. M. Isaac and S. Ember (November 8, 2016). ["For Election Day Influence, Twitter Ruled Social Media"](http://www.nytimes.com/2016/11/09/technology/for-election-day-chatter-twitter-ruled-social-media.html). [*The New York Times*](https://en.wikipedia.org/wiki/The_New_York_Times). Retrieved November 20, 2016.
6. X. Bai, F. Chen and S. Zhan. “A Study on Sentiment Computing and Classification of Sina Weibo with Word2vec”. *2014 IEEE International Congress on Big Data.* p 358-363. June 27 – July 02, 2014. *IEEE Computer Society*. Accessed: Dec. 12, 2016.