

# Sentiment Analysis of Twitter Messages Using Word2Vec

Joshua Acosta, Norissa Lamaute, Mingxiao Luo, Ezra Finkelstein, and Andreea Cotoranu  
Seidenberg School of CSIS, Pace University, Pleasantville, New York

**Abstract**— This study presents and discusses experiments in sentiment analysis of Twitter posts relating to U.S airline companies. The goal of this study is to determine whether tweets can be classified either as displaying positive, negative, or neutral sentiment. Word embeddings that are produced by word2vec are generally used to learn context and produce high-dimensional vectors in a space. These embeddings are then classified using machine-learning algorithms. The challenges of applying word2vec to tweet sentiment classification include using tweet length and syntax to determine the most accurate method to analyze Twitter posts. The highest accuracy for sentiment classification of Twitter posts was 72%. This prediction was achieved with Support Vector Machine and Logistic Regression classifiers, combined with the word2vec skip-gram training model.

**Index Terms**- Multiclass classification, natural language processing, sentiment analysis, Twitter, word embedding, word2vec

## I. INTRODUCTION

An estimated 1.96 billion people currently use social media around the world. This figure is expected to grow to at least 2.5 billion people using social media by 2018 [1]. Leaders in the social media global market include Facebook, WhatsApp, QQ, WeChat, QZone, Tumblr, Instagram, and Twitter [2]. Approximately 500 million posts are made on Twitter alone every day [6].

Through these social media applications, users freely share experiences and opinions with their network of followers. Users tend to express a variety of sentiments in their posts, therefore these posts provide invaluable insight into how the users think [3].

What is the capability of a machine to use opinion mining and sentiment analysis tools to accurately determine the feeling associated with a post? The algorithms used for sentiment analysis are not without limitation. Sentiment analysis relies heavily on pattern recognition and a basic understanding of key words. The analysis of Twitter posts presents a challenge due to the length of the posts. Every post is limited to 140 characters which makes pattern recognition more difficult. Other challenges include the inability of a machine to recognize sarcasm. This study attempts to use word2vec in order to achieve higher accuracy when compared to other algorithms that have been used for sentiment analysis. This study

integrates knowledge of stylometry, sentiment analysis, and classification algorithms and poses the following questions:

- 1) *What are the key factors to consider when classifying Twitter posts?*
- 2) *What algorithms are effective for sentiment classification?*
- 3) *What is the prediction accuracy when using word2vec for sentiment classification of Twitter posts?*

Stylometry is the literary style that differentiates a user from others. Stylometry is not only limited to user vocabulary, but also to syntax and spelling. This study analyzes a large dataset from Twitter to extract useful information for sentiment classification and an understanding of stylometry is important.

Traditional algorithms for text classification include: Naive Bayes, K-means/KNN, and Decision Trees. These algorithms have been used successfully for classification of text. However, algorithms developed more recently, such as word2vec, can build models for texts and rebuild semantic context regardless of the order of words. Word2vec works similarly to the human mind in that it uses word association to help a computer identify possible word combinations [12]. This study experiments with different algorithms to find which one yields the highest prediction accuracy.

Our study analyzes the similarities between the posts of a user in order to help verify the origin of the post. In addition to identifying similarities between posts, our study will classify their sentiments.

Sentimental analysis is a natural language processing task. By analyzing text and linguistic content, identification of the subjective information of the Twitter posts should be possible.

The objective of this study is to determine whether using the word embeddings generated by the word2vec algorithm could be used to classify sentiment. The big idea is that by using word embeddings, there will be no need to manually create features based off stylometry in order to classify sentiment accurately.

Section 2 reviews literature on sentiment analysis and the word2vec algorithm along with other effective models for sentiment analysis. Section 3 describes methodology and preprocessing of the dataset. Section 4 describes experimental results. Section 5 concludes the paper with a review of our results in comparison to the other experiments. Section 6 discusses the future work.

## II. LITERATURE REVIEW

Rong [12] describes the word2vec model and its application as well as usages and techniques for optimizing data analysis. Word2vec encompasses two models: the continuous bag-of-words model (CBOW) and the skip-gram (SG) model [12]. The SG model and the CBOW models are opposites, but they are both effective architectures for allowing the neural networks to learn words and their context [12]. The CBOW method uses the context to predict the next word and the SG model uses the word to predict the context [12]. Word2vec is used in conjunction with other algorithms in order to accurately classify sentiments.

Mikolov et al. [13] proposed two architectures for continuous vector representations of large datasets and analyzed the efficacy of the different algorithms and architectures [13]. These include Feedforward Neural Net Language Model (NNLM), Recurrent Neural Net Language Model (RNNLM), Parallel Training of Neural Networks, and new log-linear models such as CBOW and SG. Their work revealed that one could train high quality vectors on simple model architectures, such as CBOW or SG, and that due to the lower computational complexity of these architectures, higher accuracy over larger datasets could be achieved [13].

Kharde and Sonawane [8] analyzed and compared several different learning techniques and algorithms that have been tested for sentiment classification on Twitter data. [8] Their research compared the performance of a number of machine learning algorithms such as Naive Bayes, Max Entropy, Support Vector Machines as well as other lexicon-based approaches [8].

The support vector machine method, which was performed by Pang and Lee, achieved the highest accuracy. Pang and Lee's experiment yielded a classification prediction rate of 86% and a number of other algorithms all in the 80% accuracy range [3]. The main flaw of their survey is that the two different studies used different datasets [3]. The level of accuracy would be more telling if all were done using the same dataset.

Maas et al. [11] introduced a model that mixes both supervised and unsupervised sentiment components to predict document-level sentiment. They applied the model to widely tested corpora and were able to out-perform all of the other methods previously tested [11]. Traditional stop words which are commonly removed from language processing were left in because of how they help imply sentiment [11]. This also holds true for non-word tokens such as “!” and “:-)” [11]. The authors took their dataset from previous work done by Pang and Lee and used their results as benchmark. Maas et al. were able to correctly classify 88% of the test cases. [11] One important contribution of this research is the classification at the sentence level which is relevant to the classification of Twitter posts discussed in this study.

The work of Go et al. [4], had a different approach as they removed all emoticon and non-word tokens as they train their algorithms. Their research found that removing the non-word tokens allowed the classifiers to focus on other features while including them had a negative impact on two of the algorithms they tested with a negligible effect on the third [4]. After training, they applied Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms to classify Twitter sentiment with accuracy in the 80% range [4] which is in line

with the results of the work that Maas et al. conducted [11]. The best result, 83%, was achieved with the Maximum Entropy classifier when using both Unigrams and Bigrams. However, all of the results were within three percentage points so no combination of classifier and feature performed much worse [4].

Hashtags are another feature that similar to emoticons that help differentiate social media texts from other texts. Kouloumpis et al. [9] built a dataset that included hashtags (e.g. #bestfeeling, #epicfail, #news) in an effort to improve accuracy for sentiment analysis [9]. Their training set utilized what they viewed as the top positive, negative, and neutral hashtags including #iloveitwhen, #thingsilove, #success, #worst, #itsnotok, #ihate as an improved way to collect and sort training data [9]. Kouloumpis et al. tested the inclusion of emoticons against the dataset of hashtags alone and found that including emoticons showed no significant improvement versus hashtags alone [9].

Lilleberg and Yun [10] implemented word2vec for text classification. They tested word2vec against tf-idf, term frequency-inverse document frequency, and were able to show that word2vec in conjunction with tf-idf outperforms tf-idf on its own. Their findings concluded that word2vec provides complementary features to tf-idf and that the results are consistent enough to show that the combination of the two can outperform either algorithm individually [10]. Their research compared performance with and without omitting stop words and varying weighted combinations of the two. Their research also compared algorithm accuracy with respect to the number of categories it had to classify between and showed that they were extremely effective, 85% accuracy and up while restricting the test to two categories, but dropped as low as 65% when the number of categories was increased to four [10].

Wang et al. [14] explore the research and application of the sentiment analysis on Chinese reviews [14]. This study was produced to compare the efficiency of the TF-IDF algorithm, and the TF algorithm for text classification against the feature-based six-tuple vector model in conjunction with the High Adverb of Degree Count (HADC) on Chinese sentiment analysis [14]. They found that the six-tuple model and HADC weighting algorithm in conjunction succeeded in classifying between 88% and 92% of reviews accurately [14]. More research needs to be done to conclude if this is the most effective algorithm for sentiment analysis, but it is clear that this model achieves a high level of accuracy. This model has not yet been tested on English text.

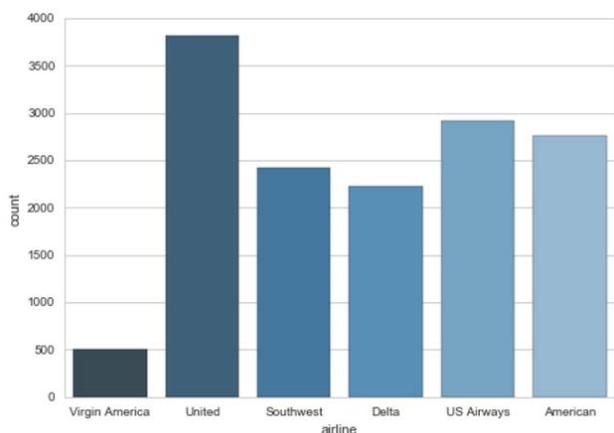
Sakar et al. [15] takes a new stance on the use of Naïve Bayes algorithm. In general, the Naïve Bayes method is not as effective as many other algorithms, however, it can be extremely accurate when used in conjunction with other methods [15]. The team proposes a two-step feature selection method which is based off of finding the most used words in the selection to create the base and then clustering to reduce feature space [15]. The study examined choosing the “important words” based on a chi-squared value and defined those words as the base by using a term based matrix [15]. This proposed method improved the performance of Naïve Bayes, reduced the feature set more effectively than the univariate chi-squared, and outperforms more traditional methods such as the greedy based

search wrapper or correlation-based feature selection (CFS) [15].

### III. METHODOLOGY

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 Tweeter posts relating to users' experiences with U.S airlines. The dataset contained 14,640 tweets and 15 attributes including the original tweet text, Twitter user-related data, and the class sentiment label. Upon downloading the dataset, it was read into a Pandas DataFrame object inside of a Jupyter notebook in order to simplify the computation step. 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. The following figures show the visual representation of the data.

Figure 1. Quantity Distribution of the Airline in Twitter Data

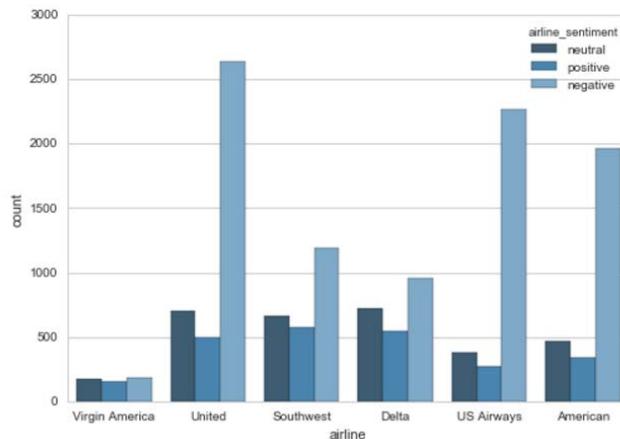


The dataset was previously labeled to indicate customer sentiment for each of the airlines. For this dataset, the user experience is described as positive, negative or neutral, as displayed in the diagram below. These classes represent the criteria by which the accuracy of the experiments were analyzed.

Our exploratory analysis showed an imbalance between the three sentiment classes with 9178 labeled as negative, 3099 labeled as neutral and 2363 labeled as positive instances. This distribution might be caused by predictions of testing instances that skewed our analysis towards negative sentiment.

The analysis also showed a higher number of negative instances for each of the six airlines (American, Delta, Southwest, United, US Airways and Virgin America) than positive and neutral which can also cause for instances to lean towards being classified negative in cases where the name or Twitter handle of the airline appears in the text. For American Airlines, United Airlines and US Airways, there are four to five times more Twitter posts classified as negative than positive which can affect the training for how the classifiers determine the sentiments of the test data or future posts. There is a discussion how this issue was handled in the Preprocessing section below.

Figure 2. Airline Sentiment Distribution



#### A. Preprocessing

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 word2vec 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.

There are primarily two types of stylistics that are used for preprocessing: stop words and tokens. Stop words refer to the most common words in a language [16], such as "a", "the", "across". Tokens refer to instances of a sequence of characters in a document grouped together as a semantic unit for processing [reference].

The NLTK's Punkt tokenizer was used in this study. The tokenizer split the words, characters, and symbols in each sentence. For example, if there are five question marks at the end of a sentence, it is possible to determine the amount of question marks that are appropriate to include in the analysis. Since Twitter limits their posts to 140 characters. No Natural Language Toolkit (NLTK) stop words, characters, or symbols were removed from the analysis. Our study was conducted under the premise that these characters are vital for the word2vec model to gain context from the words in the Twitter posts.

In order to compare the results of the different trials, two different training algorithms were instantiated: 1) one model using continuous bag-of-words (CBOW) and 2) another model using skip-gram (SG). 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. Once the

word2vec model created word embedding for each word, feature vectors were created which entailed adding all of the vectors of a word together in a Tweeter post and averaging them due to the different lengths of tweets.

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 and undersampling. Since our dataset size was limited, a random oversampler was 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 average vectors were then scaled using scikit-learn's scale function from its preprocessing library. The scaled vectors were then used to train several classifiers to see which one provided the most accurate classifications, results were recorded and are discussed in the results section.

### B. Classification

Several classifiers were implemented using the scikit-learn machine-learning library to predict the sentiment of Twitter posts. These classifiers include Naïve Bayes, Support Vector Machines and Logistic Regression. Each classifier was initiated and then the fit method was used to train the classifiers on the scaled oversampled training data. The predict method was then used to classify the test dataset, and the accuracy score method from scikit-learn's metrics library was used to determine the effectiveness of the test. Logistic Regression and Support Vector Machines classifiers performed best as reflected in Table I. The accuracy score for the classifiers were recorded and will be discussed in the results section.

### C. Similarity Test

With respect to word2vec and its implementation, similarity check is one of the assets that our study obtained. Although it is not our major concern, our researchers designed an experiment to determine the similarity between Twitter samples. In order to utilize Similarity model in genism, building up corpus based on all the tweets, vectorising through TF-IDF, using LSI model (latent semantic index) to analyze the keyword and topics of each tweet. Then a matrix form was constructed to compare the posts and output the results into a similarity matrix. The similarity matrix will also be discussed further in the results section.

## IV. EXPERIMENTAL RESULTS

### A. Results

Similar to the survey conducted by Kharde and Sonawane [8], our study compares the accuracy of Naïve Bayes, Support Vector Machines and Logistical Regression machine learning algorithms. As shown in Table I, four classifiers were trained with two different training models, SG and CBOW, embedded inside the word2vec algorithm.

The SG training model outperformed CBOW model in every classifier trained. Gaussian Naïve Bayes had an accuracy rate of 64% for CBOW, however when used with the SG model produced an accuracy rate of 69%. Bernoulli Naïve Bayes had an accuracy rate of 56% when used in conjunction with the CBOW, this accuracy rate increased to 62% when the classifier was used with the SG model. Support Vector Classifier had an accuracy rate of 70% when used with CBOW and increased to 72% when used with the SG model. Finally, Logistic Regression had an accuracy rate of 69% with CBOW and a rate of up to 72% with the SG model. All of the classifiers were instantiated with their default parameters with the exception of the Support Vector Classifier, which had its kernel parameter set to linear.

TABLE I. Accuracy Percentage by Classifier and Training Model

Classifier	Training Model	Accuracy %
Gaussian Naive Bayes	CBOW	64
Bernoulli Naive Bayes	CBOW	56
Support Vector Classifier	CBOW	70
Logistic Regression	CBOW	69
Gaussian Naive Bayes	Skip-Gram	69
Bernoulli Naive Bayes	Skip-Gram	62
Support Vector Classifier	Skip-Gram	72
Logistic Regression	Skip-Gram	72

As shown in Table II, the classification report details how the logistic regression classifier performed for each sentiment class. For the negative class, the classifier accurately predicted 75% of 2750 negative test instances correctly with a precision rate of 87% and an F1-score of 81%. For the neutral class, the classifier accurately predicted 62% of 936 neutral test instances correctly with a precision rate of 51% and an F1-score of 56%.

For the positive class, the classifier accurately predicted 70% of 706 positive test instances correctly with a precision rate of 57% and an F1-score of 63%. Clearly, the logistic regression classifier along with other training algorithms struggled to precisely classify neutral and positive tweets accurately. As shown in Figure 1, many instances that were incorrectly predicted were labeled as neutral. For example, in Figure 3, 457 out of 2750 negative tweets (17%) were predicted to be neutral while 113 out of 706 positive tweets (16%) were predicted to be neutral.

TABLE II. Classification Report of Logistic Regression Classifier

	Precision	Recall	F1-score	Support
<b>Negative</b>	0.87	0.75	0.81	2750
<b>Neutral</b>	0.51	0.62	0.56	936
<b>Positive</b>	0.57	0.70	0.63	706
<b>Average / Total</b>	0.75	0.72	0.73	4392

TABLE III. Classification Report of Support Vector Classifier

	Precision	Recall	F1-score	Support
<b>Negative</b>	0.87	0.76	0.81	2750
<b>Neutral</b>	0.52	0.63	0.57	936
<b>Positive</b>	0.59	0.70	0.64	706
<b>Average / Total</b>	0.75	0.72	0.73	4392

As shown in Table III, the classification report details how the logistic regression classifier performed for each sentiment class. For the negative class, the classifier accurately predicted 77% of 2768 negative test instances correctly with a precision rate of 87% and an F1-score of 82%. For the neutral class, the classifier accurately predicted 61% of 920 neutral test instances correctly with a precision rate of 51% and an F1-score of 56%.

For the positive class, the classifier accurately predicted 71% of 704 positive test instances correctly with a precision rate of 60% and an F1-score of 65%. Clearly Support Vector Classifier, similar to the logistic regression classifier, struggled to precisely classify neutral and positive tweets accurately. As shown in Figure 1, many instances incorrectly predicted were labeled as neutral. Similar to the other classifier, shown in Figure 2, 457 out of 2750 negative tweets (17%) were predicted to be neutral while 113 out of 706 positive tweets (16%) were predicted to be neutral.

B. Limitations

Analyzing Tweeter posts is problematic for several reasons. One of the challenges of sentiment analysis on tweets is the 140-character limit per post; after subtracting the length of a user’s twitter handle, there is not much space left to complete a proper sentence. Typically, the users of this website use abbreviated words, slang terms, or hashtags, which makes learning context difficult. Another limitation is detecting sarcasm and determining whether the user attempting to be funny or hurtful.

Another limitation, specific to this experiment, was the uneven distribution of negative tweets in comparison to the neutral and positive tweets. Although the number of training instances was increased to even out the three classes, the word2vec model continued to relearn repeated words and context from the negative class whose words and context were robust.

Figure 3. Confusion Matrix for Logistic Regression Classifier

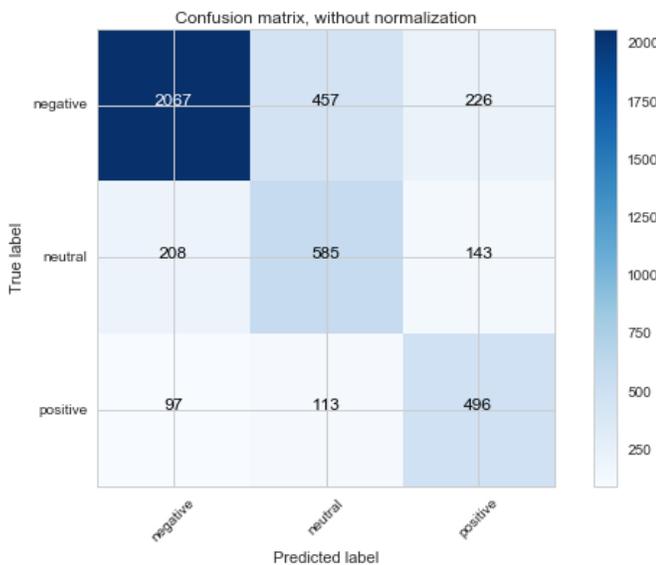


Figure 4. Confusion Matrix for Support Vector Classifier

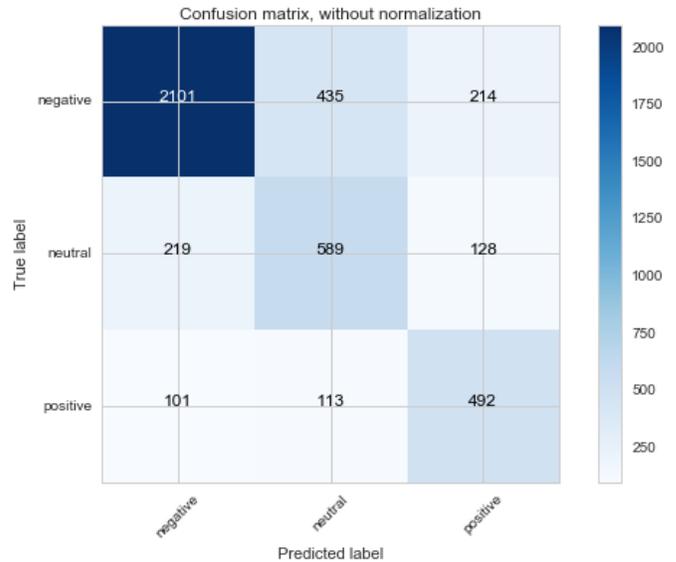


Figure 5 represents the similarity between tweet sample 0 and the rest of the samples. The similarities appear to be high, however, since twitter has 140-word limitation, it's possible that the tweets simply appear similar because the large sample size and the context of the tweets overlap.

Most of the tweets in the dataset consist of one sentence. For example, “@united yea, a refund. Your airline ruined our trip.” After removing the stop words from this sentence, the keywords are “refund”, “airline”, “ruin” and “trip”. Tweets with the same keywords will probably have high degree of similarity. Some of the keywords in dataset that appear often are words like “thanks”, “delay”, “refund”, and “time”, which are descriptors for flights but may not necessarily mean that the content of the tweets is similar. The LSI model will analyze the latent semantic inclination and generalize the topic of each posts.

In this Figure 5, the output value is determined by the cosine similarity or the similarity between sample 0 and the rest of the samples. Since TF-IDF weights cannot be negative, the output value would have to be less than 1.0 and greater than 0.

If the result value equals 1, then

$$\begin{aligned} \cos S &= 1 \\ S &= \cos^{-1} 1 = 0 \end{aligned}$$

In which S represents similarity difference. This result means the two texts have nearly the same context.

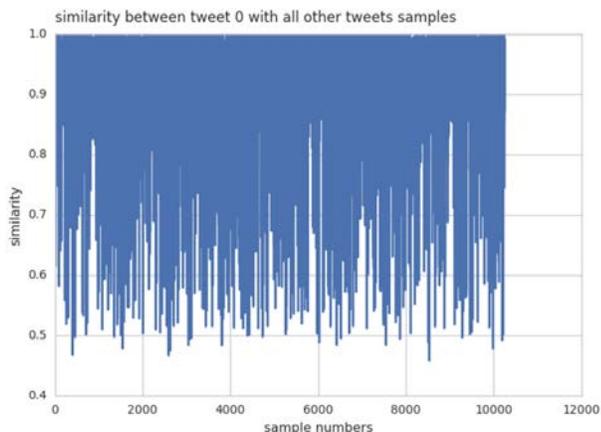
If the result value equals 0.9, then

$$\begin{aligned} \cos S &= 0.9 \\ S &= \cos^{-1} 0.9 = 0.451 \text{ rad} = 25.84 \text{ degree} \end{aligned}$$

which means that the two sentences have an angle of 25.84 degree between two term frequency vectors.

The smaller that the angle of similarity is, the more similar the two pieces of text are. For example, if the original base case was tested against itself, the angle would be 0 degrees. If the base case was tested against its opposite, the angle would be 90 degrees.

Figure 5. Similarity between tweet 0 and dataset



Social media applications like Twitter and Facebook have given end users the ability to project their voice and opinions to millions of users around the globe. It has also provided companies and governments with a seemingly endless amount of feedback regarding stakeholder 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 word2vec 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 in order to classify correctly. The dataset was acquired from Kaggle.com, which contains over 14,000 tweets about user's airline experience and their classification. Due to the imbalance among the sentiment classes, our dataset oversampled the underrepresented classes until there were equal number of training instances for each class.

The algorithms used included Naïve Bayes, Logistical Regression and Support Vector Machine algorithms in an attempt to classify over 4,000 tweets after training the word2vec algorithm with over 10,000 tweets. The highest accuracy rate yielded by a classifier was 72% using Support Vector Classifier and SG as the word2vec training model. Our next step is to continue fine-tuning the parameters of the classifiers in an attempt to increase the accuracy rates and to try different vectors.

## VI. FUTURE WORK

Our results for this experiment were not as successful as other studies that have conducted similar tests on different datasets. In the future, the team hopes to expand the research that was conducted to extend into different directions and refine the current implementation.

The team hopes to optimize the accuracy of the sentiment analysis by potentially adopting different classifiers and by analyzing other approaches that other researchers used. Also, the expansion of this work should explore more features of Word2Vec and take advantage of its vast modules to obtain more accurate results.

The second main goal of our future work is to extend the study from the similarity check. By combining the knowledge of data classification and similarities between the text, the expansion of this work could potentially move toward verification and authentication of short text.

## VII. REFERENCES

- [1] "U.S. population with a social media profile 2016 | Statista", *Statista*, 2016. [Online]. Available: <https://www.statista.com/statistics/273476/percentage-of-us-population-with-a-social-network-profile>. Accessed: Oct 7, 2016.
- [2] Most famous social network sites worldwide as of September 2016, "Global social media ranking 2016 | Statista", *Statista*, 2016. [Online]. Available: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>. Accessed: Oct 7, 2016.
- [3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," 2008. [Online]. Available: <http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf>. Accessed: Oct. 23, 2016.
- [4] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N. Project Report, Stanford 1 (2009): 12.
- [5] H. Baayen, H. van Halteren, A. Neijt, and F. Tweedie, "An experiment in authorship attribution", 6th JADT, 2002.
- [6] InternetLiveStats, "Twitter usage statistics," 2011. [Online]. Available: <http://www.internetlivestats.com/twitter-statistics/>. Accessed: Oct. 23, 2016.
- [7] J. Li, J. Monaco, L. Chen, and C. Tappert, "Authorship Authentication Using Short Messages from Social Networking Sites" Proceedings of the Information Systems Education Conference, 2014
- [8] Kharde, Vishal, and Sheetal Sonawane. "Sentiment Analysis of Twitter Data: A Survey of Techniques." arXivpreprint arXiv:1601.06971 (2016).
- [9] Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! in Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-2011). 2011.
- [10] Lilleberg, Joseph, and Yun Zhu. "Support Vector Machines and Word2vec for Text Classification with Semantic Features." *Cognitive Informatics & Cognitive Computing* (2015): n. pag. Print.
- [11] Maas, R. Daly, P. Pham, & D. Huang, (n.d.). Learning Word Vectors for Sentiment Analysis. *Proceeding HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142-150.
- [12] X, Rong, "word2vec Parameter Learning Explained", arXiv:1411.2738v4 (2016)
- [13] T. Mikolov, K. Chen, G. Corrado, J. Dean. "Efficient Estimation of Word Representations in Vector Space", arXiv:1301.3781v3, 2013.
- [14] H. Wang, L. Liu, W. Song, and J. Lu, "Feature-based sentiment analysis approach for product reviews," *Journal of Software*, vol. 9, no. 2, p. 274, Feb. 2014.
- [15] S. Sarkar, S. Goswami, A. Agrwal, J. Aktar. "A Novel Feature Selection Technique for Text Classification Using Naïve Bayes" *International Scholarly Research Notices*, vol. 2014, no. 717092, p. 1-10, Apr. 2014
- [16] A. Rajaraman. J.D. Ullman, "Data Mining". *Mining of Massive Datasets*, Cambridge, England, CUP, 2011, ch 1, pp. 1-17.