International Journal of Database Theory and Application
Vol.9, No.4 (2016), pp.289-300
http://dx.doi.org/10.14257/ijdta.2016.9.4.27

Visual and Auditory Representation of Sentiment Classified Data Using SVM

Vishal T.V.[1*], Srinidhi S.[2] and Dr. G. Muneeswari[3]

[1, 2, 3] *Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam Chennai, India*
[1]*vishalvasudevan@rocketmail.com,*
[2]*srinidhisridharan89@gmail.com,*
[3]*muneeswarig@ssn.edu.in*

Abstract

In the past few years, microblogging websites have evolved to become a source of varied kind of information. Twitter is a popular microblogging website where users create short status messages called 'tweets'. In this paper, we present a state-of-the art model trained using a support vector machine with Bag-Of-Words and TF-IDF features for each tweet. The proposed model provides a visual and an auditory representation of the sentiments that the tweets have been classified into. The results show a state-of-the art performance achieved by the model with a F1 measure of 77.47 and an accuracy of 77.93% which is better than the existing models.

Keywords*: Sentiment analysis, Tweets, SVM , TF-IDF, Bag-Of-Words, Classification, Supervised Learning*

1. Introduction

Millions of tweets are generated on a daily basis regarding various topics and issues. The easy accessibility of websites like twitter, attributes to the major shift of internet users from traditional communication tools to microblogging services. Owing to the free format of messages, people often express their opinions on various issues using twitter. As more and more users post about products, movies, or express their political and religious views, microblogging websites become valuable sources of people's expressions, opinions and sentiments. Various companies and organizations are increasingly seeking ways to exploit twitter for information on what people feel and think about their products and campaigns. There have been various techniques introduced to perform sentiment analysis on twitter data. In today's fast moving world, not everyone has the time to read the positive tweets and negative tweets about an issue raised by a person. In this paper we propose a method to classify 'tweets' into negative, positive and neutral sentiment and also update the user on this sentiment using appropriate audio sounds.

Linguistic analysis of our corpus was performed and a sentiment classifier system was built that updates the user on the sentiment of the tweet without having to read the classified label. The corpus was created using the Twitter Search API and tweets were labeled using emoticons. The tweets were initially classified using emoticons for positive and negative sentiment. The data was processed after the classification to remove the emoticons [1].

In this paper, microblogging and more particularly Twitter is selected for the following reasons:

•Microblogging platforms are used by different people to express their opinion about different topics, and, thus it is a valuable source of people's opinions.

ISSN: 2005-4270 IJDTA

•Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.

•Twitter's audience varies from regular users to celebrities, company representatives to politicians, and country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.

•Twitter's audience is represented by users from many countries. Although users from the U.S are prevailing, it is possible to collect data in different languages.

In Table 1, the mapping of various emoticons to a more general category can be visualized. If the emoticon ':)' is present, the sentiment is considered to be positive. If the emoticon ':(' is present, the sentiment is considered to be negative. The mapping allows us to reduce the different number of emoticons present in the corpus to just two. These emoticons are removed after the sentiment scores have been assigned to the tweets. Table 2 shows an example of the tweets present in the dataset that express users' opinions. The paper is organized as follows. Section 2 explores the literature survey. Section 3 describes our proposed model. Section 4 covers the results generated by our model and its efficiency. Section 5 lists the scope for future work and conclusion

Table 1. Mapping of Emoticons to a General Category

Emoticons mapped to ":)"	Emoticons mapped to ":("
:) :D :-) :) =)	:(: (:-(=(

Table 2. Examples of Twitter Posts with Expressed Users' Opinions

kippy2: i am furious with time warner and their phone promotions!
fanboy10: i hate the new movie spotlight!!
tpryan: @stellargirl I love my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right.
ashpechkam: Just got my new toy. Canon 50D. Love love love it!

2. Literature Survey

Sentiment analysis is a growing area of natural language processing and text mining. Generally, sentiment classification can be performed at four different levels: word level [2], phrase level [3], sentence level [4-5], and document level [6-9]. In the literature, there are mainly two kinds of approaches on sentiment classification: term-counting approach (lexicon-based) and machine learning approach (corpus-based). The machine learning approach can be classified as supervised and unsupervised. Sentiment analysis can be described as a classification technique where the output labels are the different sentiments that we are trying to capture. Existing supervised learning techniques can be used to classify sentiment such as Support Vector Machines (SVM), Naïve Bayes, Logistic Regression, and Artificial Neural Networks. Pang *et al*. [7] performed sentiment analysis on movie reviews by employing SVMs and Naïve Bayes classifiers which were trained on different feature sets including unigrams, to classify the sentiment of those reviews. The results show that SVMs trained on a unigram bag-of-words feature set outperforms all other approaches. Arora *et al*. [10], employed unigrams, bigrams and trigrams as features to classify restaurant reviews. SVM and Naive Bayes classifiers were used in the model which is similar to the approach taken by Pang *et al*. [7]. Zhang *et al*. [11] performed sentiment analysis on travel destination reviews with the help of bag-of-words features, SVM and Naive Bayes classifiers. Riloff *et al*. [9] extracted patterns from the corpus and used them as features along with unigrams and bigrams. The model was also implemented using SVM. Prabowo *et al*. [12] focused on MySpace comments and used Part-of-Speech tagging and N-grams as features to classify sentiment using SVM and Rule-based Classification techniques. Qu *et al*. [13] used a Bag-of-Opinions model as features as opposed to bag-of-words along with a constrained ridge classification technique. Kennedy and Inkpen [14] explore negation shifting by incorporating negation bigrams as additional features into machine learning approaches.

Often, certain specific phrases or words are the dominating indicators of the sentiment of that text. Using unsupervised learning based on such ontology to classify sentiment is another viable choice. Known opinion words can be used for classification of sentiment which was explored by Taboada *et al* [15]. Dictionary of positive and negative words were used to classify the text as positive or negative depending on the overall score. Turney[6] displayed a straight forward unsupervised learning approach for characterizing a review based on occurrences of certain phrases. Words were classified as positive or negative and how solid the assessment is by figuring the words' point wise mutual information (PMI) for their co-occurrence with a positive seed word ("excellent") and a negative seed word ("poor"). This value was called the word's semantic orientation. The method [6] accomplished 74% accuracy classifying a corpus of item reviews.

Harb *et al*. [16], performed blog classification by beginning with the two sets of seed words with positive and negative sentiment separately. Kim *et al*. [4] explore a method to perform sentiment analysis using Part-of-Speech tagging and performs sentiment classification on word level and sentence level. One of the most fundamental tasks in sentiment classification is selecting an appropriate set of features. Some of the important features are:

- Terms and their frequency: These features are individual words (unigram) and their n-grams with associated frequency counts. They are also the most common features used in traditional topic-based text classification.
- Part of speech: The part-of-speech (POS) of each word can be important too. Words of different parts of speech (POS) may be treated differently for example adjectives carry a great deal of information regarding a document’s sentiment.
- Sentiment words and phrase: Sentiment words or opinion words are words in a language that are used to express positive or negative sentiments. For example, good, awesome, and nice are positive sentiment words, and defective, poor, and risky are negative sentiment words.
- Rules of opinions: Apart from sentiment words and phrases, there are also many other expressions or language compositions that can be used to express or imply sentiments and opinions.
- Sentiment shifters: These are expressions that are used to change the sentiment orientations, *e.g.*, from positive to negative and vice versa or from negative to constructive.
- Negation words are the most important class of sentiment shifters. For example, the sentence “I don’t like this smart phone” is negative.

The task of classification can be done using various algorithms which have been elucidated below.

A Support Vector Machine builds a hyper plane or a set of hyper planes in high or infinite-dimensional space, used for classification. It is a maximum margin classifier as the algorithm generates decision boundaries that have the largest distance to the nearest training data feature vector [17]. The feature vectors that are closest to the decision boundary are called support patterns of the decision boundary. To keep the computational load reasonable, the mappings used by SVM schemes are modified to ensure that the vector products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function k(x, y). Hyperplanes are a set of points whose dot product with a vector in that space is a constant. SVMs show lesser generalization error as it is a maximum margin classifier. The performance and accuracy depends on the hyperplane selection, number of features and the type of kernel used.

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification [18]. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant then the input data will be transformed into a reduced representation set of features (also named features vector) [18]. It is a non-parametric model as stated by N.S. Altman's work on KNNs [19]. In this model, the new data point, is compared with k nearest sample data points, and the class with maximum number of nearest neighbors to the new data point is deemed as the class of the data point. It is generally used for text based analysis and sometimes fast image classification after the features of the image are extracted using Principle Component Analysis

Naive Bayes is a simple technique for constructing models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set [20]. It is a family of algorithms for training such classifiers, all based on a common principle. Naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a vegetable may be considered to be a potato if it is brown, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this vegetable is a potato, regardless of any possible correlations between the color, roundness and diameter features.

3. Visual and Auditory Representation-Methodology

In this paper, two methods are proposed to convert tweets into a corresponding audio/visual output depicting the sentiment of the tweet. The general flow of operations are illustrated below.

A. Feature Extraction

The first one is the TF-IDF approach which stands for Term Frequency-Inverse Document Frequency (Figure1). This is used to indicate how important a word is to a document (tweet) in a collection or a corpus. The TF-IDF vector for a given tweet is calculated as follows:

- The term frequency for each term is calculated by counting the number of times the term occurs in the document. The term frequency of term *t* in document *d* is given by$f_{t,d}$.
- The inverse document frequency is calculated by $log\frac{N}{n_t}$ where N is the total number of documents and n_t is the number of documents containing the term *t*.
- The TF-IDF value is then obtained by $\frac{f_{t,d}}{log\frac{N}{n_t}}$ for each word in the document. This operation is performed on each word in each tweet to obtain the vector representations for each tweet.

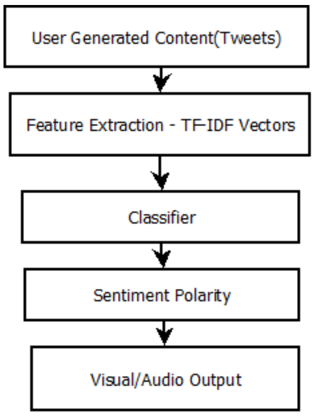

Figure 1. TF-IDF Model

The second representation is the bag-of-words representation (Figure2).

- This representation converts each word into a corresponding number, which, is nothing but the count of that word in the tweet.
- Consider the following Corpus:

Document (1): John likes to watch movies. Mary likes movies too.
Document (2): John also likes to watch football games.

- The list of unique words in the corpus is: [“John”, “likes”, “to”, “watch”, “movies”, “also”, “football”, “games”, “Mary”, “too”]
- This list has 10 distinct words. Using the indexes of the list, each document is represented as a 10-dimensional vector.

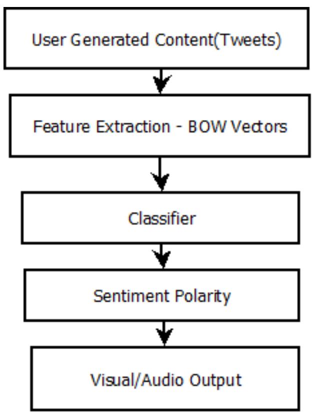

Figure 2. Bag-of-Words Model

B. Classification

The classifier used in this paper is a Support Vector Machine Classifier. Support Vector Machine is a non-probabilistic binary classifier which, when given a set of inputs belonging to two classes, outputs a hyperplane that is the optimum separation boundary between the two classes. Training samples that support the decision boundary are called support vectors. Special types of functions called kernels are used to reduce the complexity of the algorithm. When a data point is viewed as a p-dimensional vector separated by a p-1 dimensional hyperplane, the classifier is known as linear SVM classifier. In SVMs, margin is the band around the decision boundary without any training samples and maximizing the margin is important because it will reduce the test error and avoid over fitting. Any hyperplane can be represented using a set of points $\vec{x}$ as $\vec{w}.\vec{x_i} - b = 0$ where $\vec{w}$ is a normal to the hyperplane. If the data is linearly separable, two parallel hyperplanes can be drawn on either side of the decision boundary separating the two classes. Maximization of the distance between them which is the margin, is also possible. To prevent the points from going to the wrong side of the boundary the following constraint is to be considered,

$$y_i(\vec{w}.\vec{x_i} - b) \geq 1 \text{ for all } 1 \leq i \leq n \tag{1}$$

Where y_i gives the class of the data. The geometrical distance between the hyperplanes is given by $\frac{2}{\|\vec{w}\|}$. So, to maximize the margin we need to minimize $\|\vec{w}\|$ subject to

$$y_i(\vec{w}.\vec{x_i} - b) \geq 1 \text{ for } i = i \dots n \tag{2}$$

By using lagrangian multipliers we get the simplified dual form of the problem as:

$$\text{Maximize } f(c_1 \dots c_n) = \sum_{i=1}^{n} c_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i . x_j) y_j c_j \tag{3}$$

$$\text{Subject to } \sum_{i=1}^{n} c_i y_i = 0 \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \textit{ for all } i \tag{4}$$

The equation is given to a QP solver which gives out $\mathbf{c}_i$ (lagrangian multiplier). When that is non-zero, it corresponds to a support vector. The optimal hyperplane is found using the equation

$$\vec{w} = \sum_{i=1}^{n} c_i y_i \vec{x_i} \tag{5}$$

C. Visual and Auditory Tagging

The classified data is then fed to a tagger which produces outputs based on the predicted label of the tweet. This is done by hashing the predicted label with its corresponding color and sound using a dictionary. When the sentiment for a tweet is predicted, the taggers output the tweet in its respective sentiment color and also the audio. This way a person does not have to look at the tweet to know its sentiment. Even if the person wishes to look at the tweet, he can make out that the tweet is negative by viewing the color of the tweet.

Table 3. Mapping of Sentiment to Color

Sentiment	Color
Negative	Red
Neutral	Black
Positive	Green

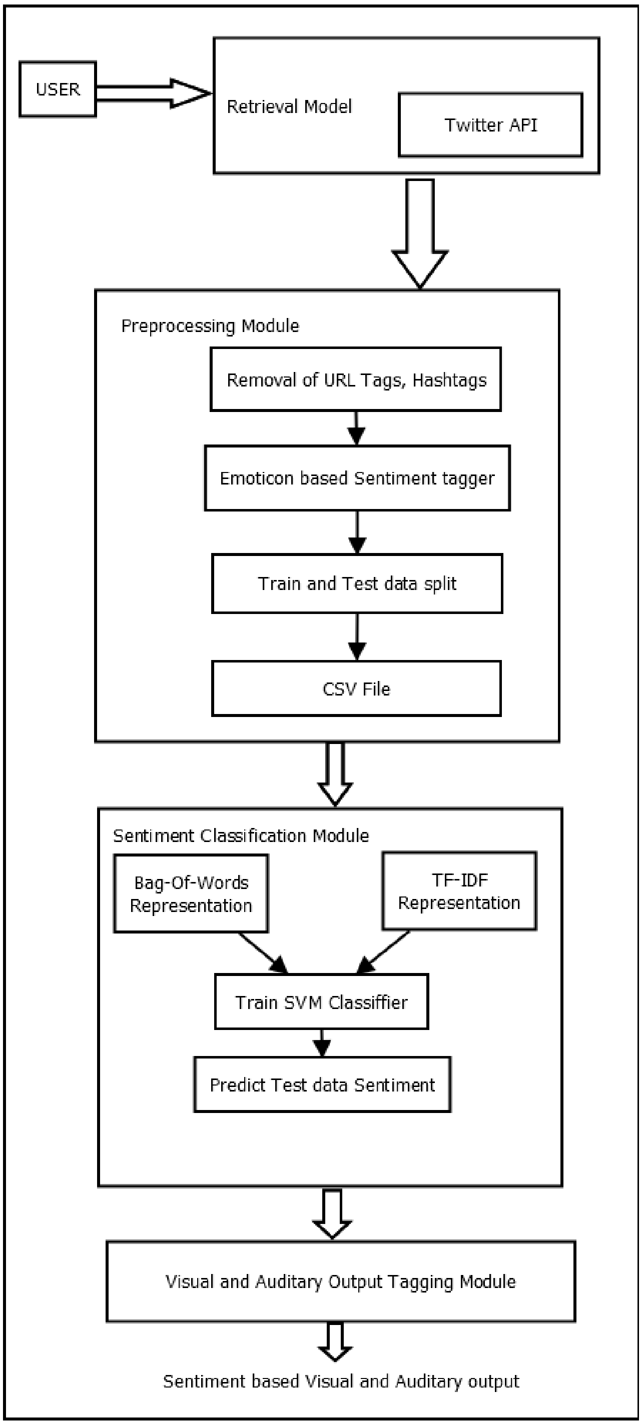

Figure 3. Preprocessing, Feature Extraction, Sentiment Classification and Output Tagging

The auditory output for each class is chosen based on the level of tone of the audio. A beep with a low amplitude is chosen for a negative sentiment and a beep with a high amplitude is chosen for a positive sentiment. In Table 3, the appropriate colors that the sentiment classes use have been assigned. Figure 3 illustrates the architecture of the proposed model. The preprocessing module corresponds to the dataset preprocessing. The emoticons are removed and the tweets are labeled with a sentiment based on the emoticons present. The sentiment classification module is responsible for feature extraction and training of the model. The visual and auditory tagging module tags the audio and color output of a tweet based on its sentiment tag.

4. Experiment and Results

For our initial set of experiments, we chose the Bag-Of-Words representation for representing the tweet as a vector. Our results show an increase in training accuracy as the number of iterations increase. The test set accuracy also increases with increase in number of iterations. The TF-IDF SVM model produces higher accuracy over the training set as well as the test set. This is due to the fact that the TF-IDF vectors do not take into account the frequent words like 'the', 'is' and so on. The Bag-Of-Words model takes these words into account and sometimes this may lead to a wrong prediction. This can be observed in Figure 4 and Figure5.

Table 4. Tweets with Corresponding Sentiment Color (Test Data)

my keyboard is all sticky from the spilt morphine
bad sleep again lacrosse in an hour
gotta get up dressed amp be productive lots to do today have a fantastic day everyone
the coffee is good this morning
The weather is okay

Table 4 clearly delineates that the classifier correctly predicts the sentiment of the tweets. The fifth tweet is classified as neutral as the term 'okay' is used in a neutral context. The auditory output is played based on the sentiment of the tweet. For testing, we used the two sounds as the audio output. This can be seen in Figure 6 and 7. The positive sentiment audio has a higher amplitude than that of the negative sentiment audio. The two sounds are easily distinguishable from one another and this provides the person with a clear distinction between the positive and negative tweet.

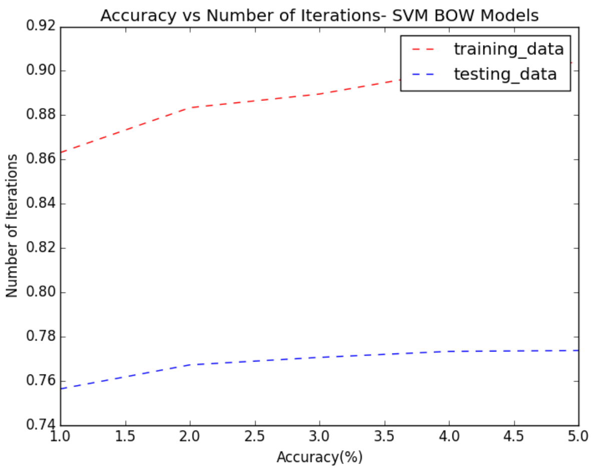

Figure 4. Accuracy vs Number of Iterations–SVM BOW Model

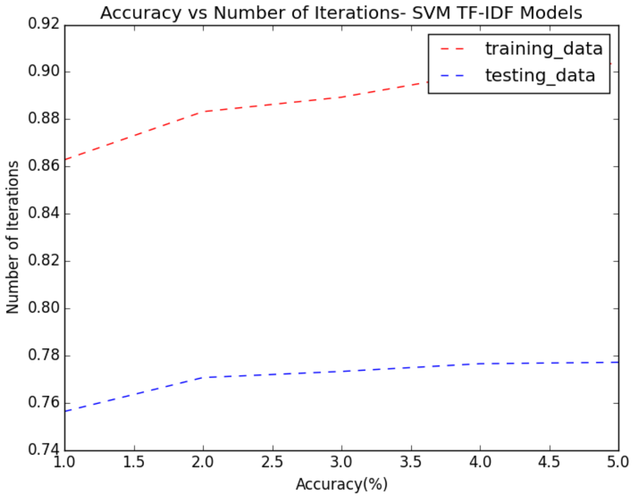

Figure 5. Accuracy vs Number of Iterations–SVM TF-IDF Model

The test data contains 320565 rows of tweets. Our Bag-Of-Words features classifier achieved an accuracy of about 77.69% on the test data. The F1 metric for the Bag-Of-Words feature model is 77.26. The TF-IDF features classifier achieves an accuracy of 77.93% and the F1 metric value is 77.47 for the same. The confusion matrix for both the models are shown in Table 5 and Table 6. From Table 5 and 6, we see that we can obtain a very high accuracy. We have also examined the impact of the dataset size on the performance of the system. To measure the performance, we use the F measure.

$$F_\beta = (1+\beta^2).\frac{precision.recall}{(\beta^2.precision)+recall} \quad (6)$$

Our results were compared to those generated by Agarwal *et al*., [22]. The maximum accuracy achieved by them is 75.39%, with a model trained on Unigrams and Senti-features. Our model surpasses the accuracy of their model by a value of 2.54%. The F1 measure attained by their model is 74.81 which is lower than the F1 measure attained by the TF-IDF which is 77.47.

Table 5. Confusion Matrix for the Bag-of-Words Model

Target Label	Predicted Label	Count (Number of tweets)
4	0	39049
4	4	121443
0	4	32449
0	0	127624

Table 6. Confusion Matrix for the TF-IDF Model

Target Label	Predicted Label	Count (Number of tweets)
4	0	38884
4	4	121608
0	4	31841
0	0	128232

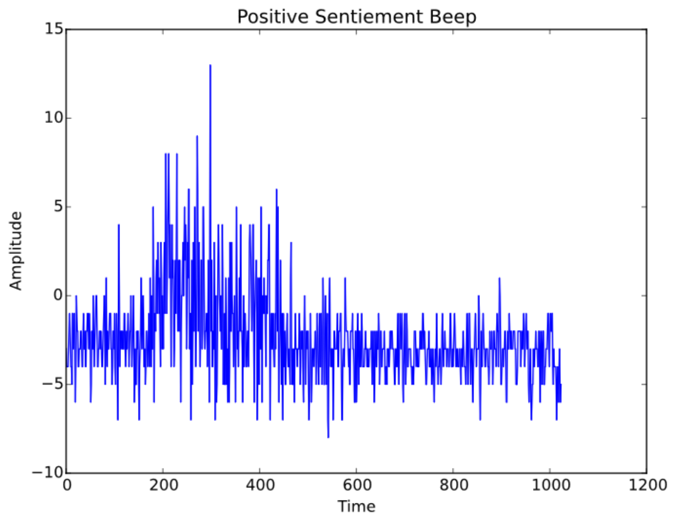

Figure 6. Positive Sentiment Audio Analysis

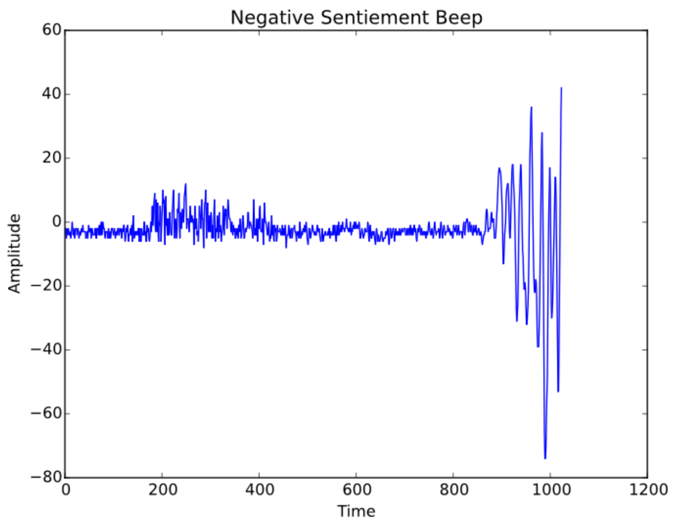

Figure 7. Negative Sentiment Audio Analysis

5. Conclusion

Microblogging has become one of the most widely used forms of communication for the people around the world. A recent research has identified that it is an online word-of-mouth branding [21]. The enormous amount of data contained in microblogging web-sites make them an attractive source of data for various machine learning tasks including opinion mining and sentiment analysis. In our paper, we have presented a method for sentiment classification and displaying these results in a way that minimizes the effort taken by the person who wishes to know the sentiment of a huge number of tweets. These tweets may be regarding product reviews or even the election campaigns. From our observations, we conclude that the TF-IDF vector representation serves as a better representation of data for our task, compared to the Bag-Of-Words representation. As the future work, implementation of a sentiment classifier can be done using deep learning methods such as Recursive Tensor Neural Networks (RNTN), to improve the accuracy of the classifier [23].

References

[1] Sentiment Analysis Dataset. Retrieved from: http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip.
[2] J. Wiebe, "LearningSubjective Adjectives from Corpora", in Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), **(2000)**, pp 735-740.
[3] J. Wiebe, T. Wilson and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis", in Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05), **(2005)**, pp 347-354.
[4] S. M. Kim and E. Hovy, "Determining the Sentiment of Opinions", in Proceedings of the COLING conference, **(2004)**.
[5] B. Liu, M. Hu and J. Cheng, "Opinion observer: analyzing and comparing opinions on the web", in Proceedings of 14th International World Wide Web Conference (WWW), **(2005)**, pp 342-351.
[6] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews", in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL), **(2002)**, pp 417-424.
[7] B. Pang, L. Lee and S. Vaidyanathan, "Thumbs up?: Sentiment Classification using machine learning techniques", in the Proceedings of the ACL-02 conference on Emperical methods in natural language processing, **(2002)**, pp 79-86.
[8] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity analysis using subjectivity summarization based on minimum cuts", in the Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics(ACL), **(2004)**.
[9] J. Wiebe, E. Riloff and S. Patwardhan, "Feature subsumption for opinion analysis", in the Proceedings of the Conference on Empirical Methods in Natural Language Processing, **(2006)**, pp 440-448.
[10] R. Arora and S. Srinivasa, "A faceted characterization of the opinion mining landscape", in the proceedings of Sixth International Conference on Communication Systems and Networks (COMSNETS), **(2014)**, pp. 1-6.
[11] Z. Zhang, Q. Ye and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches", Expert Systems with Applications, vol. 36, **(2009)**, pp. 6527-6535.
[12] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach", Journal of Informetrics, vol. 3, **(2009)**, pp. 143-157.
[13] L. Qu, R. Gemulla and G. Weikum, "A weakly supervised model for sentence-level semantic orientation analysis with multiple experts", in the proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), **(2012)**, pp 149-159.
[14] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters", Computational Intelligence, **(2005)**, pp. 110-125.
[15] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-based methods for sentiment analysis", Computational Intelligence, vol. 37, no. 2, **(2010)**, pp. 267-307.
[16] A. Harb, M. Plantie, G. Dray, M. Roche, F. Trousset and P. Poncelet, "Web Opinion Mining: How to extract opinions from blogs?", in the Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, **(2008)**, pp. 211-217.
[17] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A training algorithm for optimal margin classifiers", in the Proceedings of the fifth annual workshop on Computational learning theory - COLT, **(1992)**, pp. 144.
[18] K nearest neighbors Retrieved from: https://en.wikipedia.org/wiki/.
[19] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression", The American Statistician, **(1992)**, pp. 175-185.
[20] Naïve Bayes Classifier Retrieved from: https://en.wikipedia.org/wiki/Naive_Bayes_classifier.
[21] B. J. Jansen, M. Zhang, K. Sobel and A. Chowdury, "Micro-blogging as online word of mouth branding", in Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, **(2009)**, pp. 3859-3864.
[22] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL Workshop on Languages in Social Media, **(2011)**, pp. 30-38.
[23] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank", in Proceedings of the Conference on Emperical Methods in Natural Language Processing, **(2013)**, pp 1631-1642.

