



Sentiment Analysis of Enterprise Mashups Using Scikit and NLTK

A. M. RAJPER, S. VIGHIO, Z. HUSSAIN, A. WAGAN*

Department of Information Technology, Quaid-e-Awam University, Nawabshah, Pakistan

Received 2nd June 2012 and Revised 20th August 2012

Abstract: With the advent of internet and its related technologies, storage and use of data is increasing every second at voluminous speed. There is a need to process this huge volume of data automatically and to find out the required hidden patterns in it. The main objective of our research is to find the covert sentiments in Twitter status messages which can help companies to realize the true potential of their businesses and can also help individuals to make better decision during a purchase. The status messages (tweets) are extracted from twitter about different companies, products, and personalities and are analyzed to find out what are the opinions (positive or negative) of people about those entities? The data is extracted using twitter API. A framework is proposed which facilitates to test different techniques and find one which works best with polarity detection of twitter messages. Bigram, unigram, term frequency and inverse document frequency (TF-IDF) and other feature selection methods are used along with three machine learning algorithms: support vector machines, stochastic decent gradient, and Naïve Bayes. The work also includes performance comparison of different learning methods to analyze which kinds of techniques work best with twitter corpus.

Keywords: Sentiment analysis; Machine learning; Text mining

1. INTRODUCTION

Sentiment analysis or opinion mining can be applied to a wide variety of real life problems. There has always been a need to create systems that can automatically process the subjective contents in a document. If you want to spend holidays in Swat, Pakistan, you might be interested to review people's opinion about the desired holiday destination using a search engine which would provide you thousands of results based on your query. An alternative would be a system that provides graphical summary of the positive and negative results so that an individual can decide whether to go for the desired holiday location or not.

Sentiment analysis or opinion mining can be performed at various levels of a text document. Sentence or phrase level sentiment analysis is performed at the individual sentences of a document. The first task is to detect the subjectivity contents of a sentence if it is not a neutral sentence, then the polarity of the sentence is detected to find whether it is positive or negative. There is another similar kind of task which is carried out to find the collective polarity of a document which is known as document level sentiment classification (Liu 2011).

The work presented in (Turney 2002) evaluates the reviews regarding different domains, automobile, banks, movies and travel destinations. The reviews are collected from Epinions (see, www.epinions.com). The author has suggested an unsupervised learning algorithm approach to provide a rating of thumbs up if a review is positive or thumbs down if the review is regarded as

negative. The work in (Pang *et al.*, 2002) has proved to be valuable for our research. It is regarded as the benchmark for the researchers who are interested in opinion mining. Unlike the work in (Turney 2002), Pang *et al.* particularly focused on the reviews of movies. The authors use supervised learning algorithm to classify movie reviews into one of two classes: positive or negative. The purpose of this approach is to analyze whether the machine learning schemes: Naïve Bayes, Maximum Entropy, and SVM, work as good as they perform, in order to classify documents based on the topics. According to the paper, the accuracy is not improved by employing bigram and other features. Bag of words (unigram word presences) when used with SVM, produced best results compared to other features and classifiers.

Due to the widespread availability of user generated data over the web, huge amounts of data is available for sentiment classification. According to a survey presented in (Horriggan 2008), 81% of the Internet users have done online research on a product. Another survey (<http://www.comscore.com>) says that 73-87% online reviews significantly influence their purchase; further it reveals that consumers are willing to pay 20% to 99% more on a 5-star-rated product than a 4-star-rated product.

The work presented in this paper is mostly based on our already defended master thesis (Manzoor 2012). In this paper, we investigate the areas of natural language processing (NLP) and machine learning (ML) techniques and find their alignment with sentiment

++ Corresponding author A. M. RAJPER, Email: arajper@gmail.com,

*Department of Computer Systems Engineering, Quaid-e-Awam University, Nawabshah, Pakistan

analysis. Furthermore, we also suggest a sentiment analysis framework that can be applied on a micro-blog website like Twitter. In this regard, we develop a simple crawler that extracts tweets from Twitter and build also a corpus from tweets for sentiment classification. We implement our proposed framework in Python using NLTK (Bird *et al.*, 2009) and (Pedregosa *et al.*, 2011). Based on our analysis, we also discuss the performance evaluation of different techniques based on various metrics such as accuracy, precision, recall, and f-score.

The organization of remaining paper is as follow: Section II discusses the proposed framework and its various implementation details. The analysis and evaluation of results are presented in Section 3. Conclusions and suggestions are presented in Section 4.

2. MATERIAL AND METHODS

Proposed Framework

In this section, we briefly discuss our proposed framework; however later sections give a detailed discussion of different components of the framework as shown in (Fig.1). The framework facilitates to experiment with the corpus using various Natural Language Processing (NLP) and Machine Learning (ML) techniques in order to discover sentiment detection model that works best with our corpus. In order to create the corpus, we choose social networking website Twitter that is a platform for sharing huge amount of data. It provides an ample opportunity to exploit embedded information in order to improve business knowledge and intelligence of an organization. NLP engine removes the noisy features and extracts the features that appropriate to classify the polarity of a Tweet. While in ML module different classifiers are tested to check which one performs better with Twitter corpus.

In this section the intricacies and issues are explored regarding implementation of the framework. The implementation of the framework is done in four phases:

- Development of crawler.
- Development of NLP engine.
- Development of ML module.
- Development of sentiment detection model.

A. Development of crawler

One of the preliminary steps in sentiment analysis is to build a text corpus. A corpus is a large body of text that in our case is based on tweets. A crawler helps us to build this corpus. Crawler is a computer program that searches the website and collects the queried contents. A crawler is built on top of the Twitter API. Hence, the crawler extracts the tweets from Twitter so that the corpus can be made to manipulate for sentiment analysis (Fig.1).

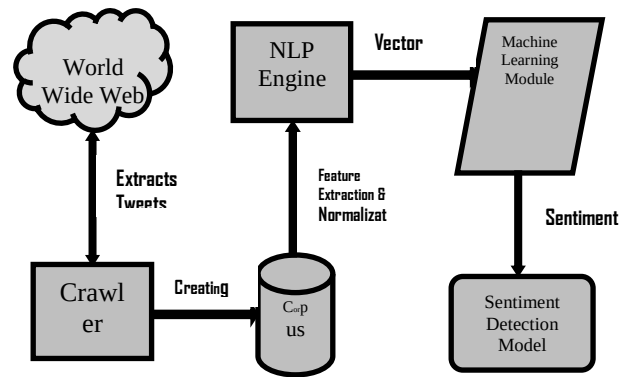


Fig.1. Proposed framework

B. Development of NLP Engine

NLP is the computational study of linguistics (Jurafsky *et al.*, 2009). Since we are dealing with human generated status messages (tweets), NLP methods are required to effectively process the text and understand the underlying semantics of tweets.

The purpose of NLP engine is two-folded, first it attempts to extract the relevant features which are necessary to detect polarity of a tweet, and second it normalizes the feature set. Normalization process helps to reduce the feature space. However, before doing any of the preceding two steps, some pre-processing of the input is required. Pre-processing involves determining the vocabulary input set. It simplifies an understanding of the language used in the input. Various NLP methods are used in order to observe which set of features and normalization techniques work better with the corpus.

C. Development of ML module

We evaluate different kinds of supervised ML techniques using different sets of parameters and methods: bag-of-words, bigrams and TF-IDF. We start with Naïve Bayes classifier as a baseline. Linear support vector machine (SVM) and stochastic decent gradient (SDG) are also employed to test which of these algorithms produce the best result.

Naïve Bayes is simple probabilistic classifier based on applying Bayes' theorem with strong independence assumption. Naïve Bayes classifier categorizes the documents based on the terms that these documents contain. It is fast and accurate when it is applied with Boolean word feature extraction.

Support vector machine is a popular classification technique that attempts to find large-margin hyper-plane. Unlike Naïve Bayes it is a non-probabilistic classification technique. The input to the SVM is vector. It is vector space based ML technique where the objective is to find a decision boundary between two classes that is maximally far from any point in the training data.

For the implementation, we use scikit learn (Pedregosa *et al.*, 2011) which is a machine learning

library written in Python. It is observed that Naïve Bayes with the highest information gain produces the best results as compared to linear SVM and SDG.

D. Development of Sentiment Detection Model

It is suggested that the better approach to develop such kind of model is to use high information gain feature with Naïve Bayes classification. It is also observed that the suitability of the techniques depend on the type of the corpus. In the context of this paper, many kinds of normalization techniques are used to reduce the feature set. Besides this different ML techniques and feature extraction method are evaluated.

In the first scenario Naïve Bayes classifier is selected using simple bag-of-words feature extraction to train our model. (Tab. 1) shows the scores for different measures.

Tab.1. Different metrics values for Naïve Bayes using Bag-of-Words method.

Metric	Positive	Negative
Accuracy	74%	
Precision	82%	69%
Recall	62%	87%
F-score	71%	77%

Performance analysis of Naïve Bayes using bag-of-words

(Fig.2) shows the ROC curve for Naïve Bayes using bag-of-words. The result of an ROC curve can be summarized in a single value which is an area under the curve. That area in this particular case is 83% as shown in Fig. 2. This value can be used to compare a model with other models; the larger the value is the better the model will be. The accuracy achieved is 74% on the data set which consists of 1 million tweets. The results show that the positive recall is 62%, this means that 38% of false negatives are in the positive class. Precision for the positive tweets is better compared to recall, that is 82 percent. It leads to 18% false positives for the positive label. Any tweet that is identified as negative is 69% likely to be correct. But any tweet that is negative is correctly identified as such with 87% recall, which means 13% false negatives in the positive class. The F-Scores do not provide useful information or any other insight. They are simply the weighted average of precision and recall.

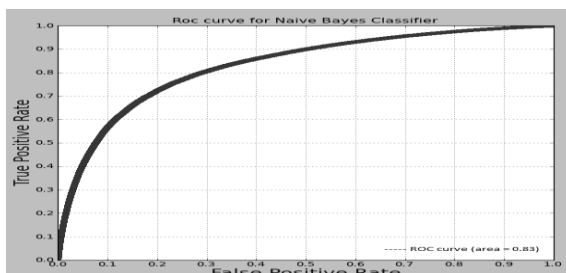


Fig.2. The Roc curve for Naïve Bayes classifier

(i) Performance analysis of linear SVM using bag-of-words

In this scenario linear SVM is used on the same data set and the same sample size using simple bag-of-words feature selection method. The results are illustrated in the (Table 2) for the same set of measures as in the previous scenario. It can be concluded from the results that the accuracy has been improved by 2%.

Table 2: Different metrics values for Linear SVM using Bag-of-Words methods.

Metric	Positive	Negative
Accuracy	76%	
Precision	77%	75%
Recall	74%	78%
F-score	76%	77%

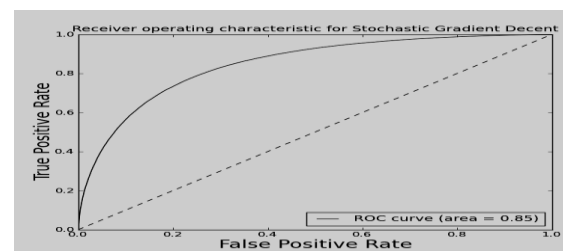
(Fig. 3) shows the ROC curve for Linear SVM, it can be seen that the area covered is 84% as compare to 83% for Naïve Bayes.

Stochastic decent gradient classifier using bag-of-words

Stochastic Decent Gradient (SGD) classifier is another ML technique which is suitable for the sparse data. (Tab. 3) shows that the accuracy is improved compared to linear SVM and Naïve Bayes.

Tab.3. SGD classifier results

Metric	Positive	Negative
Accuracy	77%	
Precision	79%	75%
Recall	73%	80%
F-score	76%	78%



(Fig.4) shows a comparative analysis among different measure of Naïve Bayes, Linear SVM and SGD classifiers using simple bag-of-words model. It can be concluded from the graph that in terms of accuracy the SGD has performed very well.

Frequency analysis of corpus

After the meticulous analysis of tweets corpus, it was realized that further normalization of the data set is inevitable. Before normalization, the feature sets for the 50 most occurring positive and negative terms contribute to 1600000 and 1800000 respectively. After normalizing, the feature set of both quantities for 50 most frequent positive and negative terms reduced to 500000 each.

Performance analysis using Naïve Bayes and highest IG features

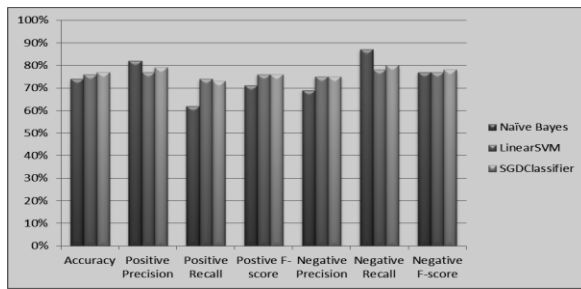


Fig. 4. A comparison of Naïve Bayes, Linear SVM and SGD classifier considering different metrics.

After eliminating low information features and making use of highest information gain features, 35000 terms are extracted. The terms are placed in a set and only those words from the tweets are extracted that appear in this set. The results are given in (Tab. 4),

Tab.4. Results after using the high information gain features

Metric	Positive	Negative
Accuracy	79%	
Precision	83%	75%
Recall	71%	86%
F-score	77%	80%

which show that the accuracy has been improved as compared to previous experiments. (Fig. 5) shows the ROC curve for Naïve Bayes classifier after eliminating

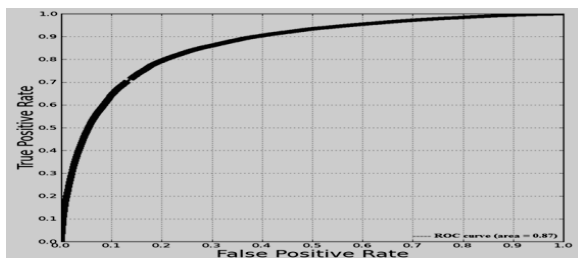


Fig. 5: ROC curve for Naïve Bayes after acquiring high IG features.

the low information features and using the high information gain features. It can be observed from the figure that the area covered under the curve is 87% which is also an improvement over the earlier results.

3. CONCLUSION AND FUTURE WORK

The main goal of this work was to apply sentiment analysis on a social networking corpus and study different NLP and ML techniques in order to see which set of techniques work best with Twitter corpus. Various techniques have been tested on Twitter corpus such as Naïve Bayes, Linear SVM, and SGD classifier with different feature-selection methods and normalization techniques. One has to take special care with Twitter corpus because of the use of informal and

misspelled words in tweets. It has been observed that after removing the noisy features and improving the feature set by consuming high information features, the results can be improved. Furthermore, it is concluded that when chi-square features selection method is used with Naïve Bayes, it produced the better results.

The suggestions are aimed at developing a framework which separates the tweets based on their geographical locations and tweets that belong to different domains. One of the potential applications would be to track the spread of contagious diseases using social media. Another potential application could be to perform sentiment analysis of individual features of a product. For instance, before buying a car, one would be interested to know the pros: interior space, seating capacity, fuel economy, driving position, and cons: Ergonomics issues, firm ride, unpleasant gas response, etc.

REFERENCES:

- Bird, S. and E. Klein, (2009) *Natural Language Processing with Python*. O'Reilly Media, Inc.
- Horrigan, J. A. (2008) Online shopping. *Pew Internet and American Life Project Report*, 36. Retrieved from http://ip-173201125199.ip.secureserver.net/_did_you_know_online_shopping.pdf
- Jurafsky D. and J. H. Martin (2009) *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd ed. Prentice-Hall, Upper Saddle River, NJ.
- Liu, B. (2011) Opinion mining and sentiment analysis. *Web Data Mining*, vol. (4): 459–526.
- Pedregosa F. (2011) Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825–2830.
- Pang B., L. Lee, and V. Shivakumar (2002) Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP '02)*, Vol. (10): 2356–2360.
- Twitter (2011). Twitter developer--Documentation. (Online) Available <https://dev.twitter.com/docs> (Accessed November 2011).
- Turney, P. D. (2002) Thumbs Up or Thumbs Down? *Semantic Orientation Applied to Unsupervised Classification of Reviews*. s.l., ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.