Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.128 – 133

RESEARCH ARTICLE

Polarity Classification Using Twitter Data

Paramjot Singh

Department of Computer Science

Amity University, India

[1] ps.takker@gmail.com

Abstract— Polarity Classification over Twitter offers different organizations a fast and effective way to monitor the feelings/emotions of general public towards their brand, business, politicians etc. A wide range of features for training polarity classifiers for Twitter datasets have been researched in recent years with varying results. In this paper, we introduce a novel approach for automatically classifying and adding semantics as additional features the polarity of Twitter messages. These messages are classified as positive or negative or neutral with respect to a query term. The paper focuses on addressing polarity classification for product features in product reviews by building semantic association between product features and polarity words. The results show that our method is encouraging.

Keywords— Corpus, tweet, analysis, sentiment, polarity, twitter

I. INTRODUCTION

Twitter has become a popular micro blogging service that has a large and rapidly growing user base where users create status messages called tweets. Users use these tweets not only as a means for updating what is on their mind, but also to express their opinion towards products, services, events and other Twitter users they are interested in. Pang et al. in [1] outline many cases where opinions expressed by Twitter users are useful in real world situations, such as product/service reviews on restaurants, electronics, hotels etc. Through performing a polarity analysis on such tweets, marketers should be able to determine the public perception of their Products and services; while consumers can know in advance what the other users think of the product/service they are interested in. Twitter, which is sometimes called a subscribe- and-publish social network, provides directed links among users' and hosts emotion-rich information across a wide set of' users and topics. Thus, it is evident that mining user opinions and polarity from Twitter will be very useful for many applications. The main objective of this research is to analyze the effectiveness of various popular classifiers and identify the more suitable classifier(s) for Twitter that could ease the process of classifying polarity in tweets.

Currently automated text analysis technology called as polarity classification is flourishing because online reviews provide an important channel for seeking out and analyzing the opinions of others. After observing enough online reviews, we found that a review or even a sentence usually involves several topics on which the holder's polarity may are not the same.

There are several challenges in the task of sentiment analysis. Firstly, we have to do subjectivity detection, i.e., selecting opinion containing sentences (Pang and Lee, 2004). Consider, for example, two sentences in a review of the city of Singapore. "Singapore's economy is heavily dependent on tourism and IT industry. It is an excellent place to live in." The first sentence is an objective or factual one and does not convey any sentiment towards Singapore. Hence this should not play any role in deciding on the polarity of the review, and should be filtered out.

Secondly, Word Sense Disambiguation (WSD), a classical NLP problem is often encountered. The problem of sentiment analysis has to grapple with thwarting, i.e., sudden deviation from positive to negative polarity, as in "The movie has a great cast, superb storyline and spectacular photography; the director has managed to make a mess of the whole thing".

Semantic relatedness measure gives the comparison of different terms or texts on the basis of their meaning or the content. For instance, it can be said that the word "computer" is semantically more related to "laptop" than "flute". Polarity Classification refers to the task of determining the overall contextual polarity of the written text. With the large range of topics discussed on Twitter, it would be very difficult to manually collect enough data to train a polarity classifier for tweets. Hence, I have used publicly available twitter datasets which are in turn obtained via distant supervision proposed in [2]. However, this dataset consist only of positive and negative tweets. For neutral tweets, we have used the publicly available neutral tweet dataset provided by [3]. We ran the machine learning classifiers Naïve Bayes, Maximum Entropy and Support Vector Machine trained on the positive and negative tweets dataset and the neutral tweets against a test set of tweets. To help visualize the utility of the Twitter-based polarity analysis tool, I have built a web application tool. This can be used by individuals and companies that may want to research polarity on any topic.

II. RELATED WORK

The research in Polarity Classification initiated with the classical machine learning algorithms like Naive Bayes, Maximum Entropy etc. using intuitive features like unigrams, bigrams, parts of speech information, position of words, adjectives etc. (Pang et. al., 2002).

However, such approaches are heavily dependent upon the given training data, and therefore can be very limited for SA due to out of vocabulary words and phrases, and different meanings of words in different contexts (Pang and Lee, 2008). Due to these problems, several methods have been investigated to use some seed words for extracting more positive and negative terms with the help of lexical resources like WordNet etc., for instance, SentiWordNet, which defines the polarity of the word along with the intensity. Sentiment analyses in Tweets are typically done in two phases: (a) identifying sentiment expressions and (b) determining the polarity of the sentiment expressed in tweets. There have been different approaches used by researchers to classify tweets and analyze sentiments and trends in Twitter. Most researchers use lexical resources and decide the sentimentality of Tweets by the presence of lexical items [7] [8]. Some other researchers combine additional features such as conjunction rules with lexical analysis to obtain more results with better accuracy [9]. This project builds on the ideas proposed in [12] where the authors classify tweets using unigram features and the classifiers are trained on data obtained using distant supervision. Read [10] shows that using emoticons (distant supervision) as labels for positive and sentiment is effective for reducing dependencies in machine learning techniques and this idea is heavily used in [12]. Pang and Lee [11] researched the performance of various machine learning techniques in the specific domain of movie reviews. However, the previous methods have not taken into account neutral tweets which lead to wrong classification and this project tries to solve this problem by including neutral tweets in the training dataset and using a novel feature vector to train the machine learning classifier and tries to classify a given tweet as positive or negative or neutral.

III. DATA

The proposed approach required the use of real time Twitter data. Data collection from Twitter involved more effort than expected and required manual labeling of posts for polarity in relation to a query. As there are only a few publicly available datasets for tweets with sufficient data needed for sentiment analysis, the Neutral- Polar-Irrelevant training dataset was manually collected by querying the Twitter Application Programming Interface (API). These queries were arbitrarily chosen from different domains to ensure variety of data. No restrictions such as language, location was made during the collection process. Thus, the collection consists of tweets in foreign languages as well. Each tweet has been labeled polar, neutral or irrelevant by an adult male fluent in English. The "irrelevant" labeled tweets are mainly Non-English tweets. The imbalanced nature of the collected dataset has been rectified as explained in a later section.

The users' privacy issues were handled as follows. Most 'tweets' on twitter are set public, and can be viewed by any person regardless of membership to twitter. The tweets which require 'following' the author in order to be accessible is termed private, and they are not be reflected on the public data stream (called the public timeline). The corpus that was used in this research was condensed from the public timeline, and hence does not reflect tweets that have been made private. Further, only the content of the tweets are collected for this research and no information pointing to the users are kept so that no means were kept, which could link a user with his/her opinion.

IV. DATA PREPROCESSING

Due to the varying and unpredictable nature of language used in tweets, it is likely that preprocessing techniques could be used to standardize certain tokens of tweets. It is highly likely that most tweets contain some form of grammatical or spelling mistakes, acronym, colloquialisms and slangs; incorporated into due to the 140 character limit imposed by
Twitter on tweets. The preprocessing process extracts the relevant content from the tweets while leaving out the irrelevant ones. The techniques applied in this paper are used commonly in information retrieval applications specifically in sentiment analysis in micro-blogging. The collected data is passed through a series of pre-processors that assist in the conversion of the message strings into the feature vector. Some of the preprocessing steps that have been carried out are explained below. This is one of the more vital steps in the entire classification process as the quality of the features/attributes that are extracted from the training dataset using the said preprocessing technique directly affects the performance of the classifiers.
Due to the varying and unpredictable nature of language used in tweets, it is likely that preprocessing techniques could be used to standardize certain tokens of tweets. It is highly likely that most tweets contain some form of grammatical or spelling mistakes, acronym, colloquialisms and slangs; incorporated into due to the 140 character limit imposed by
Twitter on tweets. The preprocessing process extracts the relevant content from the tweets while leaving out the irrelevant ones. The techniques applied in this paper are used commonly in information retrieval applications specifically in sentiment analysis in micro-blogging. The collected data is passed through a series of pre-processors that assist in the conversion of the message strings into the feature vector. Some of the preprocessing steps that have been carried out are explained below. This is one of the more vital steps in the entire classification process as the quality of the features/attributes that are extracted from the training dataset using the said preprocessing technique directly affects the performance of the classifiers.

V. MACHINE LEARNING METHODS

We tested different classifiers namely keyword-based, Naïve Bayes, Maximum Entropy and Support Vector Machines

Baseline (keyword-based)

In this approach, we used the positive and negative keyword list and for each tweet, we counted the number of positive and keywords that appear. This classifier returns the polarity of the highest count. If there is a tie, neutral polarity is returned.

Naïve Bayes

Naive Bayes is a simple model which works well on text categorization [5]. We used a multinomial Naive Bayes model. Class c∗ is assigned to tweet d, where in this formula, f represents a feature

$$c* = argmac_c P_{NB}(c|d)$$

$$P_{NB}(c|d) := \frac{(P(c)\sum_{i=1}^{m} P(f|c)^{n_i(d)})}{P(d)}$$

And ni(d) represents the count of feature fi found in tweet d. There are a total of m features. Parameters P(c) and P(f|c) are obtained through maximum likelihood estimates, and add-1 smoothing is utilized for unseen features. We used the Python based Natural Language Toolkit [18] library to train and classify using the Naïve Bayes method.

Maximum Entropy

The idea behind Maximum Entropy models is that one should prefer the most uniform models that satisfy a given constraint [20]. MaxEnt models are feature-based models. In a two class scenario, it is the same as using logistic regression to find a distribution over the classes. MaxEnt makes no independence assumptions for its features, unlike Naive Bayes. The model is represented by the following:

$$P_{ME}(c|d,\lambda) = \frac{\exp[\Sigma_i \lambda_i f_i(c,d)]}{\Sigma_{c'} \exp[\Sigma_i \lambda_i f_i(c,d)]}$$

In this formula, c is the class, d is the tweet, and λ is a weight vector. The weight vectors decide the significance of a feature in classification. A higher weight means that the feature is a strong indicator for the class. The weight vector is found by numerical optimization of the λi's so as to maximize the conditional probability. We used the Python NLTK library to train and classify using the Maximum Entropy method. For training the weights we used conjugate gradient ascent. Theoretically, MaxEnt performs better than Naive Bayes because it handles feature overlap better. However, in practice, Naïve Bayes can still perform well on a variety of problems [7].

Support Vector Machines

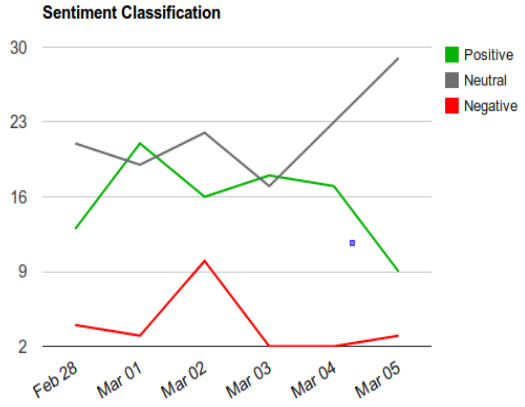
Support Vector Machines is another popular classification technique [4]. We have used libsvm [17] library with a linear kernel. My input data are two sets of vectors of size m. Each entry in the vector corresponds to the presence a feature. In the unigram feature extractor, each feature is a single word found in a tweet. If the feature is present, the value is 1, but if the feature is absent, then the value is 0. We used feature presence, as opposed to a count, so that we do not have to scale the input data, which speeds up overall processing [5].

Semantic Features for Sentiment Analysis

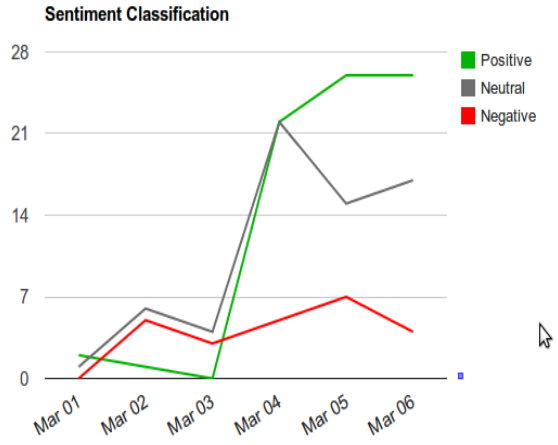
This section describes our semantic features and their incorporation into our sentiment analysis method. As mentioned earlier, the semantic concepts of entities extracted from tweets can be used to measure the overall correlation of a group of entities (e.g. all Apple products) with a given sentiment polarity. Hence adding such features to the analysis could help identifying the sentiment of tweets that contain any of the entities that such concepts represent, even if those entities never appeared in the training set (e.g. a new gadget from Apple). Semantic features refer to those semantically hidden concepts extracted from tweets [15]. An example for using semantic features for sentiment classifier training is shown in Figure 1 where the left box lists entities appeared in the training set together with their occurrence probabilities in positive and negative tweets. For example, the entities "iPad", "iPod" and "Mac Book Pro" appeared more often in tweets of positive polarity and they are all mapped to the semantic concept PRODUCT /APPLE. As a result, the tweet from the test set "Finally, I got my iPhone. What a product!" is more likely to have a positive polarity because it contains the entity "iPhone" which is also mapped to the concept PRODUCT /APPLE.

VI. RESULTS

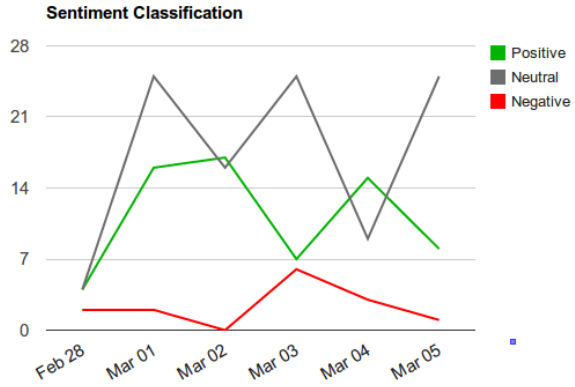
We explore the usage of unigrams and weighted unigram features and Table 4 summarizes the results.
Unigrams The unigram feature vector is the simplest way to retrieve features from a tweet. The machine learning algorithms perform average with this feature vector. One of the reasons for the average performance might be the smaller training dataset of 20000+ tweets. If one could get hold of millions of tweets and train these classifiers, the accuracy would improve substantially. Twitter API places a limit of 150 unauthorized requests per hour and hence one can download only 3600 tweets per day via the labeled tweet IDs from the publicly availably tweet ID dataset.



Virat Kohli Tweets last week Analysis



Greame Smith Tweets last week Analysis



Hashim Amla Tweets last week Analysis

Weighted Unigrams In this approach, we took advantage of the fact that it makes sense to weight the positive and negative keywords more than other words while trying to classify the sentiment of a tweet and this trick produced competitive accuracy as shown in Table 4. As expected, SVM performed the best with 80.10% accuracy and surprisingly Naïve Bayes outperformed Max Entropy by a substantial margin i.e. 78.52% to 70.42%. This is in accordance with the results shown by Pang and Lee [9]

VII. FUTURE WORK

Machine learning techniques perform well for classifying sentiment in tweets. I believe the accuracy of the system could be still improved. Below is a list of ideas we think could help the classification:-

Semantics The algorithms classify the overall sentiment of a tweet. The polarity of a tweet may depend on the perspective you are interpreting the tweet from. For example, in the tweet "Federer beats Nadal :)", the sentiment is positive for Federer and negative for Nadal. In this case, semantics may help. Using a semantic role labeler may indicate which noun is mainly associated with the verb and the classification would take place accordingly. This may allow "Nadal beats Federer :)" to be classified differently from "Federer beats Nadal :)".

Bigger Dataset The training dataset in the order of millions will cover a better range of twitter words and hence better unigram feature vector resulting in an overall improved model. This would vastly improve upon the existing classifier results.

Internationalization Currently, I focus only on English tweets but Twitter has a huge international audience. It should be possible to use my approach to classify sentiment in other languages with a language specific positive/negative keyword list.

REFERENCES

1. R. Parikh and M. Movassate, -Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques. 2009.
2. M. Wiebe, -Learning subjective adjectives from corpora, in Proc. National Conference o ArtifiCial Intelligence,2000,pp. 735-741.
3. E. Riloff and J. Wiebe, -Learning extraction patterns for subjective expressions, in Proc. Conference on Empirical methods in Natural Language Processing,2003,pp. 105-112.
4. H. Kanayama and T. Nasukawa, -Fully automatic lexicon expansion for domain-oriented sentiment analysis, in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006,pp. 355-363
5. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86, 2002.
6. J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2005.
7. Alec Go, Richa Bhayani, Lei Huang. Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford Digital Library Technologies Project, 2009.
8. Saif, H., He, Y., Alani, H.: Semantic Smoothing for Twitter Sentiment Analysis. In: Proceeding of the 10th International Semantic Web Conference (ISWC) (2011)
9. Saif, H., He, Y., Alani, H.: Alleviating Data Sparsity for Twitter Sentiment Analysis. In: Proceedings, 2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in small packages: in conjunction with WWW 2012 (2012)
10. Publicly available twitter dataset http://www.sananalytics.com/lab/twittersentiment/sanders-twitter-0.2.zip
11. D. O. Computer, C. wei Hsu, C. chung Chang, and C. jen Lin. A practical guide to support vector classification chih- wei hsu, chih-chung chang, and chih-jen lin. Technical report, 2003.
12. N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, March 2000.
13. 13. Libsvm - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
14. 14. K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999.
15. 15. B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Micro- blogging as online word of mouth branding. In CHI EA "09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pages 3859–3864, New York, NY, USA, 2009. ACM.

