Sentiment Analysis in Twitter using Machine Learning Techniques

Neethu M S
Department of Computer Science
and Engineering
College of Engineering
Trivandrum, 695016, India
msneethu29@gmail.com

Rajasree R
Department of Computer Science
and Engineering
College of Engineering
Trivandrum, 695016, India
rajasreer@gmail.com

***Abstract*—Sentiment analysis deals with identifying and classifying opinions or sentiments expressed in source text. Social media is generating a vast amount of sentiment rich data in the form of tweets, status updates, blog posts etc. Sentiment analysis of this user generated data is very useful in knowing the opinion of the crowd. Twitter sentiment analysis is difficult compared to general sentiment analysis due to the presence of slang words and misspellings. The maximum limit of characters that are allowed in Twitter is 140. Knowledge base approach and Machine learning approach are the two strategies used for analyzing sentiments from the text. In this paper, we try to analyze the twitter posts about electronic products like mobiles, laptops etc using Machine Learning approach. By doing sentiment analysis in a specific domain, it is possible to identify the effect of domain information in sentiment classification. We present a new feature vector for classifying the tweets as positive, negative and extract peoples' opinion about products.**

***Index Terms*—Twitter, Sentiment Analysis, Machine Learning Techniques**

I. Introduction

The age of Internet has changed the way people express their views. It is now done through blog posts, online discussion forums, product review websites etc. People depend upon this user generated content to a great extent. When someone wants to buy a product, they will look up its reviews online before taking a decision. The amount of user generated content is too large for a normal user to analyze. So to automate this, various sentiment analysis techniques are used.

Symbolic techniques or Knowledge base approach and Machine learning techniques are the two main techniques used in sentiment analysis. Knowledge base approach requires a large database of predefined emotions and an efficient knowledge representation for identifying sentiments. Machine learning approach makes use of a training set to develop a sentiment classifier that classifies sentiments. Since a predefined database of entire emotions is not required for machine learning approach, it is rather simpler than Knowledge base approach. In this paper, we use different machine learning techniques for classifying tweets.

Sentiment analysis is usually conducted at different levels varying from coarse level to fine level. Coarse level sentiment analysis deals with determining the sentiment of an entire document and Fine level deals with attribute level sentiment analysis. Sentence level sentiment analysis comes in between these two [1]. There are many researches on the area of sentiment analysis of user reviews. Previous researches show that the performances of sentiment classifiers are dependent on topics. Because of that we cannot say that one classifier is the best for all topics since one classifier doesnt consistently outperforms the other.

Sentiment Analysis in twitter is quite difficult due to its short length. Presence of emoticons, slang words and misspellings in tweets forced to have a preprocessing step before feature extraction. There are different feature extraction methods for collecting relevant features from text which can be applied to tweets also. But the feature extraction is to be done in two phases to extract relevant features. In the first phase, twitter specific features are extracted. Then these features are removed from the tweets to create normal text. After that, again feature extraction is done to get more features. This is the idea used in this paper to generate an efficient feature vector for analyzing twitter sentiment. Since no standard dataset is available for twitter posts of electronic devices, we created a dataset by collecting tweets for a certain period of time.

By doing sentiment analysis on a specific domain, it is possible to identify the influence of domain information in choosing a feature vector. Different classifiers are used to do the classification to find out their influence in this particular domain with this particular feature vector.

II. Related Work

There are two basic methodologies to detect sentiments from text. They are Symbolic techniques and Machine Learning techniques [2]. The next two sections deal with these techniques.

A. Symbolic Techniques

Much of the research in unsupervised sentiment classification using symbolic techniques makes use of available lexical resources. Turney [3] used bag-of-words approach for sentiment analysis. In that approach, relationships between the individual words are not considered and a document is represented as a mere collection of words. To determine the overall sentiment, sentiments of every word is determined and those values are combined with some aggregation functions. He found the polarity of a review based on the average semantic orientation of tuples extracted from the review where tuples are phrases having adjectives or adverbs. He found the semantic orientation of tuples using the search engine Altavista.

Kamps et al. [4] used the lexical database WordNet [5] to determine the emotional content of a word along different dimensions. They developed a distance metric on WordNet and determined the semantic orientation of adjectives. WordNet database consists of words connected by synonym relations. Baroni et al. [6] developed a system using word space model formalism that overcomes the difficulty in lexical substitution task. It represents the local context of a word along with its overall distribution. Balahur et al. [7] introduced EmotiNet, a conceptual representation of text that stores the structure and the semantics of real events for a specific domain. Emotinet used the concept of Finite State Automata to identify the emotional responses triggered by actions. One of the participant of SemEval 2007 Task No. 14 [8] used coarse grained and fine grained approaches to identify sentiments in news headlines. In coarse grained approach, they performed binary classification of emotions and in fine grained approach they classified emotions into different levels.

Knowledge base approach is found to be difficult due to the requirement of a huge lexical database. Since social network generates huge amount of data every second, sometimes larger than the size of available lexical database, sentiment analysis became tedious and erroneous.

B. Machine Learning Techniques

Machine Learning techniques use a training set and a test set for classification. Training set contains input feature vectors and their corresponding class labels. Using this training set, a classification model is developed which tries to classify the input feature vectors into corresponding class labels. Then a test set is used to validate the model by predicting the class labels of unseen feature vectors.

A number of machine learning techniques like Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM) are used to classify reviews [9]. Some of the features that can be used for sentiment classification are Term Presence, Term Frequency, negation, n-grams and Part-of-Speech [1]. These features can be used to find out the semantic orientation of words, phrases, sentences and that of documents. Semantic orientation is the polarity which may be either positive or negative.

Domingos et al. [10] found that Naive Bayes works well for certain problems with highly dependent features. This is surprising as the basic assumption of Naive Bayes is that the features are independent. Zhen Niu et al. [11] introduced a new model in which efficient approaches are used for feature selection, weight computation and classification. The new model is based on Bayesian algorithm. Here weights of the classifier are adjusted by making use of representative feature and unique feature. 'Representative feature' is the information that represents a class and 'Unique feature' is the information that helps in distinguishing classes. Using those weights, they calculated the probability of each classification and thus improved the Bayesian algorithm.

Barbosa et al. [12] designed a 2-step automatic sentiment analysis method for classifying tweets. They used a noisy training set to reduce the labeling effort in developing classifiers. Firstly, they classified tweets into subjective and objective tweets. After that, subjective tweets are classified as positive and negative tweets. Celikyilmaz et al. [13] developed a pronunciation based word clustering method for normalizing noisy tweets. In pronunciation based word clustering, words having similar pronunciation are clustered and assigned common tokens. They also used text processing techniques like assigning similar tokens for numbers, html links, user identifiers, and target organization names for normalization. After doing normalization, they used probabilistic models to identify polarity lexicons. They performed classification using the BoosTexter classifier with these polarity lexicons as features and obtained a reduced error rate.

Wu et al. [14] proposed a influence probability model for twitter sentiment analysis. If @username is found in the body of a tweet, it is influencing action and it contributes to influencing probability. Any tweet that begins with @username is a retweet that represents an influenced action and it contributes to influenced probability. They observed that there is a strong correlation between these probabilities.

Pak et al. [15] created a twitter corpus by automatically collecting tweets using Twitter API and automatically annotating those using emoticons. Using that corpus, they built a sentiment classifier based on the multinomial Naive Bayes classifier that uses N-gram and POS-tags as features. In that method, there is a chance of error since emotions of tweets in training set are labeled solely based on the polarity of emoticons. The training set is also less efficient since it contains only tweets having emoticons.

Xia et al. [16] used an ensemble framework for sentiment classification. Ensemble framework is obtained by combining various feature sets and classification techniques. In that work, they used two types of feature sets and three base classifiers to form the ensemble framework. Two types of feature sets are created using Part-of-speech information and Word-relations. Naive Bayes, Maximum Entropy and Support Vector Machines are selected as base classifiers. They applied different ensemble methods like Fixed combination, Weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

Certain attempts are made by some researches to identify the public opinion about movies, news etc from the twitter posts. V.M. Kiran et al. [17] utilized the information from other publicly available databases like IMDB and Blippr after proper modifications to aid twitter sentiment analysis in movie domain.

III. Proposed Solution

A dataset is created using twitter posts of electronic products. Tweets are short messages with full of slang words and misspellings. So we perform a sentence level sentiment analysis. This is done in three phases. In first phase preprocessing is done. Then a feature vector is created using relevant features. Finally using different classifiers, tweets are classified into positive and negative classes. Based on the number of tweets in each class, the final sentiment is derived.

A. Creation of a Dataset

TABLE I
Statistics of the Dataset Used

Dataset	Positive	Negative	Total
Training	500	500	1000
Test	100	100	200

Since standard twitter dataset is not available for electronic products domain, we created a new dataset by collecting tweets over a period of time ranging from April 2013 to May 2013. Tweets are collected automatically using Twitter API and they are manually annotated as positive or negative. A dataset is created by taking 600 positive tweets and 600 negative tweets. Table 1 shows how dataset is split into training set and test set.

B. Preprocessing of Tweets

Keyword extraction is difficult in twitter due to misspellings and slang words. So to avoid this, a preprocessing step is performed before feature extraction. Preprocessing steps include removing url, avoiding misspellings and slang words. Misspellings are avoided by replacing repeated characters with 2 occurrences. Slang words contribute much to the emotion of a tweet. So they can't be simply removed. Therefore a slang word dictionary is maintained to replace slang words occurring in tweets with their associated meanings. Domain information contributes much to the formation of slang word dictionary.

C. Creation of Feature Vector

Feature extraction is done in two steps. In the first step, twitter specific features are extracted. Hashtags and emoticons are the relevant twitter specific features. Emoticons can be positive or negative. So they are given different weights. Positive emoticons are given a weight of '1' and negative emoticons are given a weight of '-1'. There may be positive and negative hashtags. Therefore the count of positive hashtags and negative hashtags are added as two separate features in the feature vector.

Twitter specific features may not be present in all tweets. So a further feature extraction is to be done to obtain other features. After extracting twitter specific features, they are removed from the tweets. Tweets can be then considered as simple text. Then using unigram approach, tweets are represented as a collection of words. In unigrams, a tweet is represented by its keywords. We maintain a negative keyword list, positive keyword list and a list of different words that represent negation. Counts of positive and negative keywords in tweets are used as two different features in the feature vector. Presence of negation contribute much to the sentiment. So their presence is also added as a relevant feature.

All keywords cannot be treated equally in the presence of multiple positive and negative keywords. Therefore a special keyword is selected from all the tweets. In the case of tweets having only positive keywords or only negative keywords, a search is done to identify a keyword having relevant part of speech. A relevant part of speech is adjective, adverb or verb. Such a relevant part of speech is defined based on their relevance in determining sentiment. Keywords that are adjective, adverb or verb shows more emotion than others. If a relevant part of speech can be determined for a keyword, then that is taken as special keyword. Otherwise a keyword is selected randomly from the available keywords as special keyword. If both positive and negative keywords are present in a tweet, we select any keyword having relevant part of speech. If relevant part of speech is present for both positive and negative keywords, none of them is chosen. Special keyword feature is given a weight of '1' if it is positive and '-1' if it is negative and '0' in its absence. Part of speech feature is given a value of '1' if it is relevant and '0' otherwise.

Thus feature vector is composed of 8 relevant features. The 8 features used are part of speech (pos) tag, special keyword, presence of negation, emoticon, number of positive keywords, number of negative keywords, number of positive hash tags and number of negative hash tags.

D. Sentiment Classification

After creating a feature vector, classification is done using Naive Bayes, Support Vector Machine, Maximum Entropy and Ensemble classifiers and their performances are compared.

IV. Classification Techniques

There are different types of classifiers that are generally used for text classification which can be also used for twitter sentiment classification.

A. Nave Bayes Classifier

Nave Bayes Classifier makes use of all the features in the feature vector and analyzes them individually as they are equally independent of each other. The conditional probability for Naive Bayes can be defined as

$$P(X|y_j) = \Pi_{i=1}^{m} P(x_i|y_j) \quad (1)$$

'X' is the feature vector defined as X=$\{x_1,x_2,....x_m\}$ and y_j is the class label. Here, in our work there are different independent features like emoticons, emotional keyword, count of positive and negative keywords, and count of positive and negative hash tags which are effectively utilized by Naive Bayes classifier for classification. Nave Bayes does not consider the relationships between features. So it cannot utilize the relationships between part of speech tag, emotional keyword and negation.

B. SVM Classifier

SVM Classifier uses large margin for classification. It separates the tweets using a hyper plane. SVM uses the a discriminative function defined as

$$g(X) = w^T \phi(X) + b \tag{2}$$

'X' is the feature vector, 'w' is the weights vector and 'b' is the bias vector. $\phi()$ is the non linear mapping from input space to high dimensional feature space. 'w' and 'b' are learned automatically on the training set. Here we used a linear kernel for classification. It maintains a wide gap between two classes.

C. Maximum Entropy Classifier

In Maximum Entropy Classifier, no assumptions are taken regarding the relationship between features. This classifier always tries to maximize the entropy of the system by estimating the conditional distribution of the class label. The conditional distribution is defined as

$$P_\lambda(y|X) = 1/Z(X)exp\left\{\sum_i \lambda_i f_i(X,y)\right\} \tag{3}$$

'X' is the feature vector and 'y' is the class label. Z(X) is the normalization factor and λ_i is the weight coefficient. f_i(X,y) is the feature function which is defined as

$$f_i(X,y) = \begin{cases} 1, & \text{X=x}_i \text{ and } y = y_i \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In our feature vector, the relationships between part of speech tag, emotional keyword and negation are utilized effectively for classification.

D. Ensemble classifier

Ensemble classifiers can be of different types. They try to make use of the features of all the base classifiers to do the best classification. The base classifiers used here are Nave Bayes, Maximum entropy and SVM. Here an ensemble classifier is generated by voting rule. The classifier will classify based on the output of majority of classifiers.

V. Evaluation

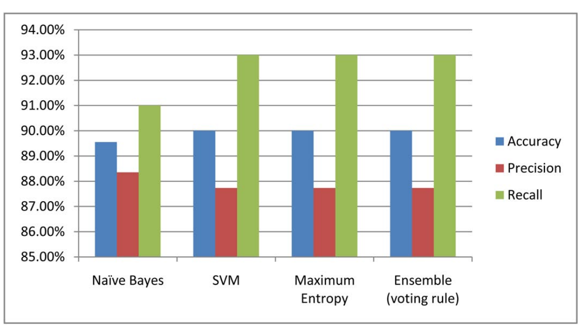

Fig. 1. Performance of Different classifiers in Twitter Sentiment Analysis

Using Twitter API, tweets related to products are collected. A dataset is created using 1200 twitter posts of electronic products. Dataset is split in to a training set of 1000 tweets and a test set of 200 tweets. We used Stanford postagger[1] for extracting part of speech tag from tweets.

Since we have selected product domain, there is no need of analyzing subjective and objective tweets separately. To identify the quality of product, both of these qualities contribute similarly. This shows how context or domain information affects sentiment analysis. These classifiers are tested using Matlab simulator. We used three types of basic classifiers (SVM, Nave Bayes, Maximum Entropy) and ensemble classifier for sentiment classification. SVM and Naive Bayes classifiers are implemented using Matlab built in functions. Maximum Entropy classifier is implemented using MaxEnt software[2]. Performance of these classifiers is shown in Fig. 1. All these classifiers have almost similar performance.

Naive Bayes has better precision compared to the other three classifiers, but slightly lower accuracy and recall. SVM, Maximum Entropy Classifier and Ensemble classifiers have similar accuracy, precision and recall. They obtained an accuracy of 90% whereas NaiveBayes has 89.5%. This shows the quality of the feature vector selected for the product domain. This feature vector aids in better sentiment analysis despite of the classifier selected.

VI. Conclusions

There are different Symbolic and Machine Learning techniques to identify sentiments from text. Machine Learning techniques are simpler and efficient than Symbolic techniques. These techniques can be applied for twitter sentiment analysis. There are certain issues while dealing with identifying emotional keyword from tweets having multiple keywords. It is also difficult to handle misspellings and slang words. To deal with these issues, an efficient feature vector is created by doing feature extraction in two steps after proper preprocessing. In the first step, twitter specific features are extracted and added to the feature vector. After that, these features are removed from tweets and again feature extraction is done as if it is done on normal text. These features are also added to the feature vector. Classification accuracy of the feature vector is tested using different classifiers like Nave Bayes, SVM, Maximum Entropy and Ensemble classifiers. All these classifiers have almost similar accuracy for the new feature vector. This feature vector performs well for electronic products domain.

References

[1] Y. Mejova, "Sentiment analysis: An overview," *Comprehensive exam paper, available on http://www. cs. uiowa. edu/~ ymejova/publications/CompsYelenaMejova. pdf [2010-02-03]*, 2009.

[2] E. Boiy, P. Hens, K. Deschacht, and M.-F. Moens, "Automatic sentiment analysis in on-line text," in *Proceedings of the 11th International Conference on Electronic Publishing*, pp. 349–360, 2007.

[3] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424, Association for Computational Linguistics, 2002.

[1] nlp.stanford.edu/software/stanford-postagger-2013-04-04.zip

[2] http://www.cs.grinnell.edu/ weinman/code/

[4] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using wordnet to measure semantic orientations of adjectives," 2004.

[5] C. Fellbaum, "Wordnet: An electronic lexical database (language, speech, and communication)," 1998.

[6] D. Pucci, M. Baroni, F. Cutugno, and A. Lenci, "Unsupervised lexical substitution with a word space model," in *Proceedings of EVALITA workshop, 11th Congress of Italian Association for Artificial Intelligence*, Citeseer, 2009.

[7] A. Balahur, J. M. Hermida, and A. Montoyo, "Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model," *Affective Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 88–101, 2012.

[8] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 70–74, Association for Computational Linguistics, 2007.

[9] G. Vinodhini and R. Chandrasekaran, "Sentiment analysis and opinion mining: A survey," *International Journal*, vol. 2, no. 6, 2012.

[10] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[11] Z. Niu, Z. Yin, and X. Kong, "Sentiment classification for microblog by machine learning," in *Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on*, pp. 286–289, IEEE, 2012.

[12] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44, Association for Computational Linguistics, 2010.

[13] A. Celikyilmaz, D. Hakkani-Tur, and J. Feng, "Probabilistic model-based sentiment analysis of twitter messages," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pp. 79–84, IEEE, 2010.

[14] Y. Wu and F. Ren, "Learning sentimental influence in twitter," in *Future Computer Sciences and Application (ICFCSA), 2011 International Conference on*, pp. 119–122, IEEE, 2011.

[15] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of LREC*, vol. 2010, 2010.

[16] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences: an International Journal*, vol. 181, no. 6, pp. 1138–1152, 2011.

[17] V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of twitter," in *Analyzing Microtext Workshop, AAAI*, 2011.