Summarizing Public Opinions in Tweets

Nibir Nayan Bora

School of Computer Engineering, KIIT University
Bhubaneswar, Orissa, India
nibirbora@gmail.com

Abstract. The objective of Sentiment Analysis is to identify any clue of positive or negative emotions in a piece of text reflective of the authors opinions on a subject. When performed on large aggregations of user generated content, Sentiment Analysis may be helpful in extracting public opinions. We use Twitter for this purpose and build a classifier which classifies a set of tweets. Often, Machine Learning techniques are applied to Sentiment Classification, which requires a labeled training set of considerable size. We introduce the approach of using words with sentiment value as noisy label in a distant supervised learning environment. We created a training set of such Tweets and used it to train a Naive Bayes Classifier. We test the accuracy of our classifier using a hand labeled training set. Finally, we check if applying a combination of minimum word frequency threshold and Categorical Proportional Difference as the Feature Selection method enhances the accuracy.

Keywords: Sentiment classification, supervised learning, Twitter

1 Introduction

A simple search for 'Kindle 2[1]' on Google[2] returns over 700 million results, mostly consisting of product descriptions and user reviews. Any user would be overwhelmed with such huge amounts of content, making it almost impossible to scan through all of it to find what is crucial. With the rise in the number of social networking, blogging and microblogging websites, and the ease with which a user can submit information on these sites, the internet today has become an unmanageable accumulation of user generated content. Users, differing in social, political and cultural background, share their personal opinion on various subjects, discuss current issues and write about events in their life. What is much needed in such a situation is a system which can extract significant information from such large sets of data; that is, give us more aggregated results.

Sentiment Analysis or Opinion Mining, the study of computationally determining whether a given piece of text is indicative of positive or negative sentiment, is one way of summarizing large aggregations of text. Sentiment classification, when performed on user generated textual content, find many applications. Knowing how users feel about a product or service can help in business

[1] http://amazon.com/kindle
[2] http://google.com

decisions for corporates. Political parties and social organizations can collect feedback about their programs and legislation. Artists, musicians and other entertainment icons can reach out to their fans and assess the quality of their work. Broadly, it can serve as an automatic polling system, relieving any manual intervention.

Twitter[3], one of the most popular microblogging websites among the internet community, serves as a good platform for sentiment analysis because of its large user base from different sociocultural zones. Users update short messages (called Tweets) within a 140 character limit on Twitter. It contains huge number of tweets, with millions added each day, which can be easily collected through its APIs (Application Program Interface), making it convenient to build a large training set. Usually, machine learning techniques are applied to sentiment classification, in which a classifier is required to be trained on a labeled training set. This is called supervised learning. However, owing to its nature and the number of tweets that can be collected, it is a challenging task to manually label a training set of such magnitude.

In this paper, we introduce the novel approach of labeling large sets of tweets using sentiment suggestive words as noisy labels. Using this method we create an annotated dataset of 1.5 million tweets, which is used to train a machine learning classifier. The accuracy of the classifier is evaluated and compared to previous similar techniques. Further, a combination of minimum word frequency threshold and categorical proportional difference is discussed as the Feature Selection method enhances the accuracy of the classifier. We also show the aggregated results of our classifier when tried on a few search queries.

The rest of the paper is organized as follows. Related works are discussed in section 2. Our approach of labeling and sentiment classification is described in section 3. The results and discussions of the experiment are presented in section 4. Finally we conclude about our work in section 5.

2 Related Work

A number of prior work has been directed towards sentiment classification on blog posts [5], reviews [9,14] and tweets [3,7,2]. However, blog posts and reviews differ from tweets, first, because of their size, and second, because tweets follow an entirely different language model, which does not ensure correctness or consistency of grammar and spelling. Because of Twitter's 140 character limit, tweets are more likely to contain spelling and grammatical errors. Users, sometimes, strips off letters from words (e.g. *whr*) to fit their message within this limit. Twitter users also use a lot of Internet slang and abbreviations (e.g. *OMG*, *ASAP*, etc.) and emoticons. Emoticons, also known as *smileys* are glyphs constructed using the characters available on a standard keyboard, representing a facial expression of an emotion.

Various approaches of sentiment classification has be studied in the past years. Initially focused on lexicon based methods (Das and Chen [1]) and un-

[3] http://twitter.com

supervised hand made algorithms (Turney [13]), the study later moved on to supervised machine learning techniques (Pang et al. [9]) and unsupervised clustering algorithms (Yessenov and Misailović [14]). Recent researches in sentiment analysis has incorporated Natural Language Processing techniques as well as Feature Selection methods to further enhance the accuracy of the classifiers. While most of these researches were concerned with classification at document level, Mishne and de Rijke [5] analyzed global mood levels at an aggregate level. This is quite similar to our study, where we classify tweets at document level, but we discuss how such an classification could be applied to a set of tweets to determine aggregated results.

Our approach of using sentiment suggestive words is similar to the use of emoticons as noisy labels, first introduced by Read [11]. A noisy label is an element within the piece of text which provides information about the class to which the text belongs. This approach was later used by Go et al. [3], and Pak and Paroubek [7] on a Twitter corpus. In such an approach, a set of emoticons is manually classified as bearing positive or negative sentiment. The occurrence of any of these emoticons in a tweet causes it to be labeled as belonging to the corresponding class. Go et al. [3] created an annotated training set using this technique. However, using emoticons as noisy labels, prevents their use while classifying new tweets. We address this drawback by using sentiment suggestive words as noisy label and hence permitting the use of emoticons during classification. Duurkoop [2] used a similar approach, however, he used hashtags instead of emoticons as noisy label in his training data.

Feature selection, used in order to improve the performance of the classifier, is a method to select a portion of the feature set, generated by the trainer, which is most likely to serve in classification. Keefe and Koprinska [6] evaluated a range of feature selection methods using both Naive Bayes classifier and Support Vector Machine on a movie review corpus. In their study, they found categorical proportional difference (CPD) to outperform other feature selection methods. CPD has also being studied by Simeon and Hilderman [12] in their study on text categorization. Yessenov and Misailović [14] considered various feature extraction and feature reduction methods in their model. They observed an increase in the performance of their classifier while considering only words that appear most frequently in the corpus. Pak and Paroubek [7] used a similar approach, an entropy based method, for feature selection. However, these feature selection methods haven't been studied on a twitter corpus. We evaluate CPD along with a minimum word frequency threshold as the feature selection method for our classifier.

3 The Experiment

3.1 Our Approach

Our study was directed towards building a classifier which could classify tweets at document level, i.e. each tweet individually. The classifier would then be utilized to analyze a collection of tweets to arrive at aggregated results. A training

Table 1. List of positive and negative sentiment words

Positive sentiment words	Negative sentiment words
amazed, amused, attracted, cheerful, delighted, elated, excited, festive, funny, hilarious, joyful, lively, loving, overjoyed, passion, pleasant, pleased, pleasure, thrilled, wonderful	annoyed, ashamed, awful, defeated, depressed, disappointed, discouraged, displeased, embarrassed, furious, gloomy, greedy, guilty, hurt, lonely, mad, miserable, shocked, unhappy, upset

set of 1.5 million tweets was built by collecting tweets using the Twitter Search API[4]. This annotated training set was then used to train a Naive Bayes classifier, treating each tweet as a bag-of-unigrams feature. In the bag-of-unigrams model, each tweet is considered as an unordered collection of words, disregarding grammar and their order of occurrence. We used words with sentiment value (e.g. *cheerful*, *shocked*, *disappointed* etc.) as noisy labels to automatically label the training set. The feature extraction process follows a series of cleaning and preprocessing steps on the tweet before tokenizing it. An attempt was made to increase the accuracy of the classifier by using two feature selection methods, namely, minimum word frequency threshold and categorical proportional difference. To test the accuracy of the classifier, a hand labeled test set was used.

3.2 The Corpus

A set of 40 words was prepared, each indicating certain sentiment value. Each of these words were then manually categorized as being positive or negative. Most of these words describe a certain mood or emotion, or are in the past tense verb form, most likely to be used by an author to express his or her feelings. e.g. *ashamed* in *"I was ashamed of what I did"*. This list has been intuitively selected and is in no way exhaustive. The complete list of positive and negative sentiment words, used in our experiments, is given in Table 1.

A tweet is labeled as positive if it contained any of the positive sentiment words, or negative if it contained any of the negative sentiment word. For example, a tweet containing the word 'thrilled' will be labeled as a positive tweet, whereas a tweet containing 'annoyed' will be labeled as a negative tweet. Table 2 shows a few tweets labeled in this manner. In disputable situations when a tweet contains both a positive and a negative sentiment word, the tweet is discarded.

The twitter search API allows to retrieve the most recent tweets based on a search query. However, it returns only 100 tweets per page and up to 15 such pages. The API also limits the number of requests per hour from an IP (Internet Protocol) address. Tweets were collected using the positive and negative sentiment words as the query term. Considering the hourly limits and allowing enough time for the requests to yield unique tweets, our system sent out a search request for each word every 30 minutes.

[4] https://dev.twitter.com

Table 2. Example of a few labeled tweets

Tweet	Label	Matched sentiment word
`Arrived in Basel. Brilliant sunny weather. I'll go to the botanical gardens first. It's a wonderful place to think and write.`	positive	wonderful
`I'm really unhappy with myself at this point and i'm seriously at a breaking point and i'm on the verge of relapse i'm trying so hard`	negative	unhappy
`Toy Story 3 is hilarious. I love the scene where Ken is modelling for Barbie! Fab stuff :-)`	positive	hilarious
`i should really eat something, but i'm just not a fan when it's this hot and miserable. :/`	negative	miserable

A large accumulation of tweets was built during the period from June 24, 2011 to July 5, 2011. All retweets were removed as they could cause an unwanted redundancy in the training set. Retweets are tweets that are posted by a user by copying another user's tweet. They are marked by the presence of the characters 'RT' in the tweet. The matching sentiment word found in each tweet was also removed. This was done so that the classifier is not trained with a bias for the set of sentiment words used for labeling. The tweets were then labeled according to the category in which the matching sentiment word falls. This way, a training set of 1.5 million (1,464,638, precisely) tweets consisting of 668,975 positive tweets and 795,661 negative tweets was created.

3.3 The Naive Bayes Classifier

Naive Bayes classifier is a probabilistic model which estimates the probability that a tweet belongs to a specific class (positive or negative class, in this experiment). Naive Bayes has been used because of its low computational overhead and ease of use, yet performing well in many Natural Language Processing tasks, as indicated by Lake [4]. In general, the probability that a tweet T, belongs to class C, can be calculated using Bayes Theorem, as follows

$$P(T|C) = \prod_i P(w_i|C) \tag{1}$$

where $P(w_i|C)$ is the probability of the feature w_i occurring in class C. Finally, the class C is assigned to the tweet T, whichever yields the maximum $P(C|T)$.

3.4 Unigram Feature Extractor

The following preprocessing steps were applied to the Tweets:

- Remove URLs (e.g. *http://bit.ly/aDkhG*) and user mentions. User mention is a way to tag a user in a tweet. It is represented by a '@' symbol followed by the username (e.g. *@nibirbora*).
- Replace emoticons with a token representing the emotion expressed by it. Table 3 is a list of few emoticons and their meanings.

Table 3. Emoticons and their meanings

Emoticons	Meaning
`>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)`	smile
`>:D :-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3`	laugh
`:'( ;*( :_(`	cry
`>:[ :-( :( :-c :c :-< :< :-[ :[ :{`	frown
`>;] ;-) ;) *-) *) ;-] ;] ;D`	wink
`>:o >:O :-O :O`	surprise
`D:< >:( >:-C >:C >:O D-:< >:-( :-@ :@ ;( '_' D< :L`	angry

- Tokenize words by splitting the tweet at spaces and punctuation marks.
- Remove stop words. Stop words are relatively common words used in a language. e.g. *the*, *is*, *at*, *which*, *on*.
- Replace words with repeated letters. Users sometimes arbitrarily repeat certain letters in a word to put more emphasis on it (e.g. *happppppyyyyyy*). We designed an algorithm which replaces a word with any letter occurring more than twice with two words, one in which the repeated letter occures once and twice in the second. For example, the word *happppppyyyyyy* will be replaced with four words - *hapy*, *hapyy*, *happy*, *happyy*.
- Stem words to their roots, so that grammatical inflections are removed. We use the Porter Stemming algorithm [10] for this purpose.

3.5 Feature Selection

We use Categorical Proportional Difference along with a minimum word frequency threshold as our feature selection method. This two step feature selection process is discussed next.

Minimum Word Frequency Threshold. In the first step, all features with frequency below a minimum threshold frequency are remove from the feature set. The threshold is set as a percentage calculated on the maximum frequency of any feature in the feature set. The accuracy of the classifier is checked at various minimum threshold percentages.

Table 4. Effect of feature selection on the feature set. (Initial feature size = 319,719)

CPD thresholds	Minimum word frequency threshold			
	1.000%	0.100%	0.010%	0.001%
0.000	1,357	6,616	27,487	319,719
0.125	883	4,553	21,042	296,861
0.250	533	2,902	15,417	287,159
0.375	258	1,643	10,409	270,971
0.500	128	903	6,919	26,615

Categorical Proportional Difference. Categorical Proportional Difference (CPD), a measure of how equal two numbers are, can be used to find the features that occur mostly in either one of positive or negative class of tweets. The positive and negative frequencies of a feature are used to calculate it's CPD, by an equation suggested by Keefe and Koprinska [6] as follows:

$$CPD = \frac{|Positive_f - Negative_f|}{Positive_f + Negative_f} \tag{2}$$

If a feature is prevalent in either positive tweets or in negative tweets then it's CPD will be close to one, whereas if it occurs almost evenly in both positive and negative tweets then its CPD will be close to zero. A high CPD indicates that the feature is worth considering for classification. For example if the word "wife" appears in exactly as many positive tweets as negative tweets then finding the word "wife" would not contribute in classifying new tweet and its CPD score will be zero. Conversely, if the word "birthday" appears in only positive tweets then finding the word "birthday" in a new tweet would give us a good clue that the tweet is positive, and it would have a CPD score of one.

CPD is used for feature selection by removing any features whose CPD score is less than some threshold value. The accuracy of the classifier was checked at various threshold value.

Table 4 shows the effect of the feature selection process on the feature set. The values indicate the feature set size after applying feature selection. The corresponding accuracies for each of these values are presented later.

3.6 Test Set

To determine the accuracy of the classifier, a test set was created. A portion of the training set was randomly selecting, from which, tweets which do not suggest any positive or negative sentiment were manually removed. The remaining tweets were then hand labeled. The final test set consisted of 198 positive sentiment tweets and 204 negative sentiment tweets.

3.7 Aggregate Classifier

As pointed out previously, the document level classifier is eventually used to find aggregated results. This is done by feeding a set of tweets to the classifier, which then classifies each document separately, and calculates the number of documents marked as positive and negative as a percentage over the total number of tweets in the set.

4 Results & Discussion

The classifier showed an accuracy of 77.61%, on the hand labeled test set, when tested without any feature selection method. However, it reached a maximum accuracy of 83.33% when feature selection methods were incorporated. This value

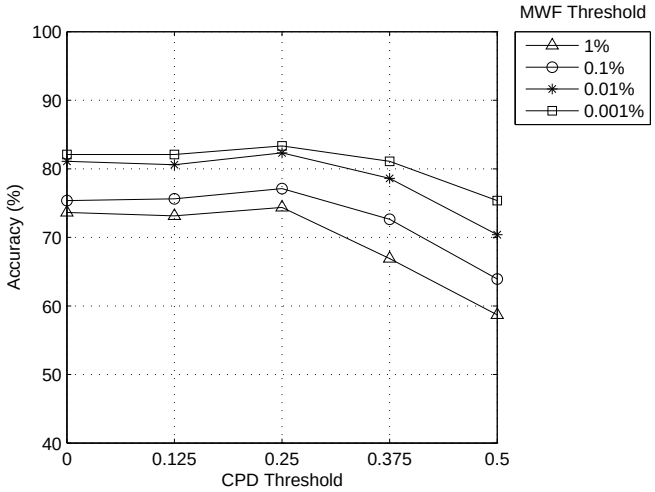

Fig. 1. Accuracy of the classifier at different minimum word frequency thresholds and CPD thresholds.

Table 5. Accuracy values (in percentage) at different minimum word frequency thresholds and CPD thresholds.

CPD thresholds	Minimum word frequency threshold 1.000%	0.100%	0.010%	0.001%
0.000	73.63	75.37	81.09	82.08
0.125	73.13	75.62	80.59	82.08
0.250	74.37	77.11	82.33	**83.33**
0.375	66.91	72.63	78.60	81.09
0.500	58.70	63.93	70.39	75.37

was achieved when the minimum word frequency threshold was set to 0.001% and the CPD threshold to 0.25, reducing the size of the feature set to 287,159, which is about 90% of the original (319,719 features). It was noticed that our feature selection method considerably increased the accuracy of the classifier. The accuracy at various minimum word frequency (MWF) thresholds and CPD thresholds are presented in Figure 1 and Table 5.

Clearly, the accuracy decreases gradually with an increase in the MWF threshold, whereas, over the range of CPD thresholds, it shows a peak at CPD value 0.25. Beyond this value the graph fall sharply, which could be attributed to the fact that more useful features are removed from the feature space. At lower CPD values (≤ 0.25), it was observed that the accuracy remained within a 10% range with change in the MWF threshold. Since considering features with frequency above a minimum threshold frequency did not contribute much to the accuracy of the classifier, we tested our classifier with CPD alone as the feature selection method. This yielded a maximum accuracy of 83.08%, almost similar to that which was achieved considering a minimum threshold frequency. Once again, this accuracy was achieved at the CPD threshold of 0.25, assuring it to

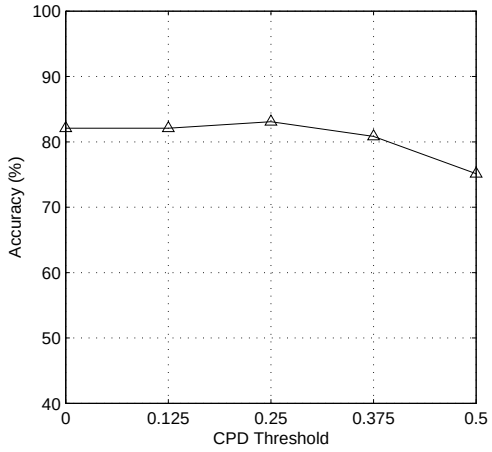

Fig. 2. Accuracy of the classifier at different CPD thresholds. (Without minimum word frequency threshold)

Table 6. Accuracy values (in %) at different CPD thresholds. (Without minimum word frequency threshold)

CPD thresholds	0.000	0.125	0.250	0.375	0.500
Accuracy	82.08	82.08	**83.08**	80.84	75.12

be an effective choice for feature reduction. The results are shown in Figure 2 and Table 6.

The maximum accuracy of the classifier was similar to that reported by Pang et al. [9], i.e. 81% and Go et al. [3], i.e. 81.34%. However, it was less than that accounted in Pang & Lee's [8] later study, i.e. 86.4%, since we did not perform any subjectivity analysis. One notable benefit of our approach is that we consider emoticons in the classification process. Emoticons, which often are a strong indicator of sentiment in short texts, were ignored by Go et al. [3] while classifying tweets. It was observed that the best accuracy of our classifier is slightly higher than those reported by other studies involving Naive Bayes classification. This may be because our test set was biased towards our training set, owing to our method of test set generation.

While our accuracy matched that of Go et al. [3], who used emoticons as noisy labels, it was quite higher than that of Duurkoop [2] (around 70%) who used hashtags, indicating that sentiment suggestive words are more effective as noisy labels than hashtags.

Yessenov and Misailović [14] accounted for an increase in the performance of their classifier while considering only words that appear most frequently in the corpus. This is contrary to our observations where we found that limiting

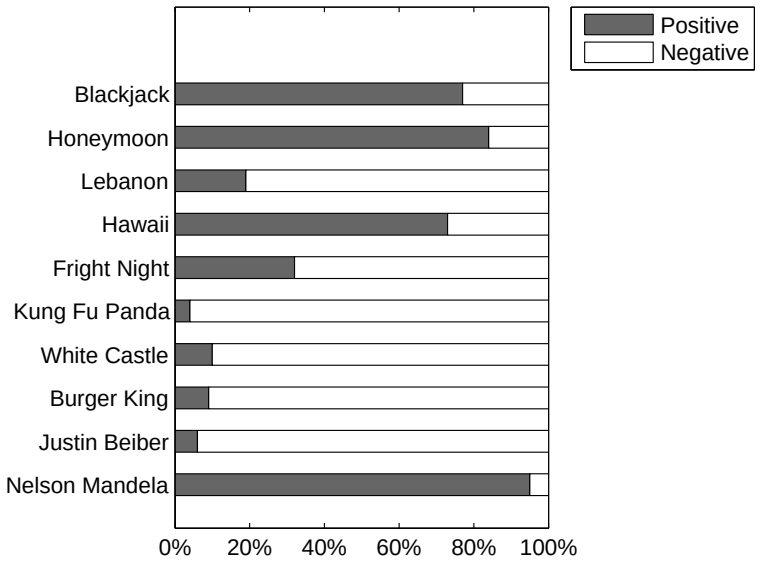

Fig. 3. Results of the classifier on a set of tweets. (Percentage positive and percentage negative)

Table 7. Results of the classifier on a set of tweets. (Values in number of tweets)

Query term	Positive	Negative	Total
Nelson Mandela	63	3	66
Justin Beiber	2	30	32
Burger King	5	49	54
White Castle	9	79	88
Kung Fu Panda	3	76	79
Fright Night	29	63	92
Hawaii	66	25	91
Lebanon	11	48	59
Honeymoon	76	15	91
Blackjack	44	13	57

the feature set with a minimum word frequency threshold does not contribute to increasing the accuracy of the classifier.

Figure 3 shows a few results of the aggregate classifier. A set of tweets related to the query terms shown in Table 7 were fed to the classifier. Table 7 also shows the number of total tweets analyzed and the number among them that were classified as positive and negative.

5 Conclusion

We built a sentiment classification tool which could accurately find the polarity (positive or negative) of a tweet, and can be used to analyze a collection of tweets to find aggregated results. We showed that sentiment suggestive words can be effectively used as noisy label for a Twitter corpus, and using this technique built a training set of 1.5 million tweets. This corpus was used to train a Naive Bayes classifier, who's accuracy was measured using a hand labeled test set. The maximum accuracy achieved was 83.33%, which is comparable to prior related studies. We also studied the use of a combination of minimum word frequency threshold and Categorical Proportional Difference as feature selection method. It was found that while CPD successfully increased the efficiency of the classifier (reaching a peak at CPD value 0.25), setting a minimum word frequency threshold did not.

Future work will include building a classification tool that can classify various emotions and not just identify positive and negative sentiment. Such a tool may be used to create a user interface which is reflective of the user's mood. We will also consider finding a way to eliminate tweets from the training set that does not indicate any sentiment value and also perform grammatical semantic analysis on the tweets to possibly increase the accuracy of classification.

Acknowledgment

The author would like to thank Prachet Bhuyan of the School of Computer Engineering, KIIT University, Bhubaneswar, Orissa for his valuable comments and suggestions.

References

1. Das, S., Chen, M.: Yahoo! for amazon: Extracting market sentiment from stock message boards. In: In Asia Pacific Finance Association Annual Conf. (APFA). (2001)
2. Duurkoop, J.: Real-Time Happiness. Bachelor thesis, University of Amsterdam, Faculty of Science, Science Park 904, 1098 XH, Amsterdam (2010)
3. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing (2009) 1–6
4. Lake, T.: Twitter sentiment analysis. Technical report, Western Michigan University, Kalamazoo (2011)
5. Mishne, G., de Rijke, M.: Capturing global mood levels using blog posts. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, AAAI (2006) 145–152
6. O'Keefe, T., Koprinska, I.: Feature selection and weighting methods in sentiment analysis. In: Proceedings of the 14th Australasian Document Computing Symposium. (2009)
7. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Seventh International Conference on Language Resources and Evaluation, LREC 2010, La Valletta, Malta, May 19-21, 2010, Proceedings, Valletta, Malta, European Language Resources Association (ELRA) (2010) 1320–1326
8. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Scott, D., Daelemans, W., Walker, M.A., eds.: ACL, ACL (2004) 271–278
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10. EMNLP '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 79–86
10. Porter, M.: An algorithm for suffix stripping. Program **14**(3) (1980) 130–137
11. Read, J.: Using emoticons to reduce dependency in machine learning techniques for sentiment classification. Proceedings of the ACL Student Research Workshop on ACL 05 **43**(June) (2005) 43
12. Simeon, M., Hilderman, R.J.: Categorical proportional difference: A feature selection method for text categorization. In Roddick, J.F., Li, J., Christen, P., Kennedy, P.J., eds.: AusDM. Volume 87 of CRPIT., Australian Computer Society (2008) 201–208
13. Turney, P.D.: 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. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 417–424
14. Yessenov, K., Misailović, S.: Sentiment analysis of movie review comments. 6.863 Spring 2009 final project, CSAIL, MIT (2009)