Investigating Active Learning Techniques for Document Level Sentiment Classification of Tweets

Abstract

Active Learning is a novel method to automatically select the useful samples from the unlabeled data in such a way that the overall classification performance improves. The creation of training examples otherwise involves huge costs and efforts and hence, is a major constraint in machine learning techniques. In this paper, we investigate the effectiveness of active learning for sentiment classification. The algorithm iterates over a number of times, and in each iteration uncertain samples from the unlabeled data are added to the training data. Our experiments on a benchmark dataset of Tweets show an overall accuracy of 83.95% which is an increment of 6.75% from the base level accuracy. The approach is very general and not domain-specific, and thus is scalable, domain-adaptable and easy to implement for a wide variety of problems.

1 Introduction

Recent decade has seen an upsurge in the social networking and e-commerce sector witnessing an enormous growth in the volume of data flowing out of media networks, which can be used by public and private organizations alike to gain valuable insights into the likes and dislikes of individuals. New forms of communication, such as microblogging, Tweets, status, reviews and text messaging have emerged and become ubiquitous. These messages often share an opinion about products, situations, books, movies, places etc. The abundance of social contents creates an opportunity to study the public sentiments which can lead to better formulation of policies and products.

Social media text, however, is beset with a large number of typos and ad-hoc abbreviations that limit the accuracies of text processing algorithms, applied subsequently. How to handle such challenges so as to automatically mine and understand the sentiments conveyed through such informal messages over social media has only very recently been the subject of the research (**?**; **?**; Barbosa and Feng, 2010; Pak and Paroubek, 2010; Kouloumpis et al., 2011). Sentiment analysis (Pang and Lee, 2008), which essentially refers to the task of finding the opinions of authors about specific entities has wide applications. Some of theses include identifying situational awareness during mass emergencies (Verma et al., 2011) or content analysis of tweets during h1n1 outbreak (Chew and Eysenbach, 2010).

Sentiment Analysis of Tweets, a distinct but related field in the domain of social media analytics, is a research area that has drawn immense interest from the scientific community for its applications in the fields of commerce (Jansen et al., 2009), disaster management (Verma et al., 2011) and health (Chew and Eysenbach, 2010). Machine learning techniques are more popular because of its easy adaptability to new applications and domains. But, the lack of sufficient training data has always been an issue for the machine learning experiments. The creation of labelled corpus is often very expensive and requires a lot of efforts.

Active Learning (Settles, 2010) plays an important role in this context as in this method, the learner itself choose the examples to learn further and hence the number of examples to learn a concept is less. These examples are considered to be important for the classifier to learn as they were uncertain in their predictions. Active Learning optimizes the control of model growth and it greatly reduces the time and costs involved in preparing the data as well as the model. Without AL, the knowledge base models grow with the increase of size of the already built data set. The strength of active learning lies in the fact that

it selects only a subset of tokens which are useful for a given classifier. It has wide applications in various tasks, particularly in the scenarios of resource-scarce environment. The motivation for active learning stems from the fundamental assumptions that are commonly shared in the Natural Language Processing (NLP) community. The assumptions being that obtaining high quality annotated data for supervised learning is expensive and time-consuming.

This paper presents a method for document level sentiment classification of Tweets based on Support Vector Machine (SVM) (Cortes and Vapnik, 1995), and investigates the effectiveness of active learning techniques (Settles, 2010) to improve the classification accuracy with minimal resources. Prior works show the use of active learning techniques for sentiment classification, particularly to deal with the imbalanced dataset(Li et al., 2012) and to study the problems of domain adaptation (Li et al., 2013).

2 Methodology and Approach

At first we develop a method based on supervised SVM (Cortes and Vapnik, 1995) for sentiment classification. We treat this as the base learner that classifies Tweets into *positive* and *negative* classes. Thereafter we propose an active learning method (discussed in Section 4) to automatically select the most informative instances from the unlabelled data for the classifier's training. This is performed in such a way that the performance of the target test data improves.

2.1 Support Vector Machines (SVMs)

Support Vector Machines (Cortes and Vapnik, 1995) are supervised machine learning models with associated learning algorithms that analyze data and recognize patterns.

Consider $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, which represents the training data for the two-class problem where $y_k \in \{+1, -1\}$ represents the class associated with $\mathbf{x}_k$ and $\mathbf{x}_k \in R^D$ is the feature vector corresponding to the k-th sample in the training set. The aim of the SVM is to learn a linear hyperplane that divides the negative examples from the positive examples such that the separation between the two classes is maximal. The equation of this hyperplane may be obtained as follows: $(\mathbf{w}.\mathbf{x}) + b = 0 \quad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}$.

In our work we make use of the SVM implementation as available with the LibLinear [1] model ((Fan et al., 2008)). LibLinear (Fan et al., 2008) is an efficient implementation which is found to be most suitable for the given task because it is optimized to handle very large feature sets. It is an efficient linear classifier that has been optimized for data with millions of instances and very large feature sets.

2.2 Features used for Sentiment Classification

Like any other classification problems appropriate features play a very important role for SVM learning. Being motivated by the prior works on sentiment classification of Tweets, for example, (Mohammad et al., 2013), we implement a the following set of features that captures various syntactic and semantic properties of Tweets.

1. N-Grams:

   An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The concept of extracting n-grams is similar to looking at a large piece of text through a window, such that the size of window is fixed.

   In our learning framework, we consider all the word-level n grams of sizes 1, 2, 3 and 4 as features. The feature is binary valued where 1 indicates the presence of that particular n-gram in the Tweets (training/test set) and 0 represents the absence of it. An illustrative example showing how a feature vector is computed is shown in Table 1. We used *N-Gram Statistics Package (Test-NSP)* (Banerjee and Pedersen, 2003) for extracting n-grams from the training and test datasets [2].

2. Part-of-Speech (PoS) information: Part-of-Speech (PoS) information may be useful in classifying the sentiments. We used *CMU ARK Twitter Part-of-Speech Tagger* [3] for extracting PoS information of the tokens present in the Tweets. The CMU ARK tagger identifies tokens in a piece of text and classifies them into 25 PoS tags, including Twitter specific tags such as '*hashtag*', '*at-mention*', '*emoticon*', '*URL*' etc. in addition

[1] www.csie.ntu.edu.tw/~cjlin/liblinear

[2] http://ngram.sourceforge.net/

[3] http://www.ark.cs.cmu.edu/TweetNLP

S.No.	Tweet	Unigrams					Bigrams				Feature Vector
		this	it	is	great	bad	this is	is great	is bad	it is	
1	this is great	1	0	1	1	0	1	1	0	0	101101100
2	this is bad	1	0	1	0	1	1	0	1	0	101011010
3	it is great	0	1	1	1	0	0	1	0	1	011100101

Table 1: Illustrative example: Ngram features

to the grammatical tags such as noun, adjective, verb etc.

Examples of some of the PoS tags used by the tagger are: N (Nouns: book, ccd), A (Adjective: good, colorful), ! (Interjection: haha, Lol, yo), E (Emoticon: :) :(:D), # (Hashtags: #worldcup), @ (At-mentions: @PMOIndia), U (URL: http://goo.gl/et4fHxS) etc.

We use each PoS tag as a feature and determine the feature value as the total number of occurrences of each PoS tag in the Tweet.

3. Lexicon features:

We use two automatically created sentiment lexicons, namely *NRC Hashtag Sentiment Lexicon* and the *Sentiment140 Lexicon* (Mohammad et al., 2013) to extract the features. For each token instance sentiment scores are calculated following the procedure as describe below:

NRC Hastag Sentiment Lexicon [4]: In tweets there exists few specific words that are marked with the hashtags (#) in order to indicate the topic or sentiment expressed in it. The hash-tagged emotion words like *joy*, *sadness*, *angry* and *surprised* are good indicators to designate that the tweet as a whole expresses the same emotion (Mohammad, 2012). Based on this idea, (Mohammad et al., 2013) defined this lexicon containing entries for 54,129 uni-grams, 316,531 bigrams and 308,808 non-contiguous pairs denoting word-sentiment associations. In this lexicon, the individual scores of the token have been calculated based on the number of tweets in which these token co-occurred with positive or negative hashtags such as #good, #excellent, #terrible etc.

Sentiment140 Lexicon [5]: We use the lexicon (Mohammad et al., 2013), created from the sentiment140 corpus (Go et al., 2009). In this lexicon, the individual scores of the token have been calculated based on the number of Tweets in which these tokens co-occurred with positive or negative emoticons such as :), :(, :D.

For each token of the Tweet, following features are extracted:

(a) Number of tokens in the tweet with $score(w) > 0$.

(b) The total score = $\sum_{w \in tweet} score(w)$.

(c) The maximal score of any token in the tweet = $\max_{w \in tweet} score(w)$.

(d) The score of the last positive token ($score(w) > 0$) in the tweet.

4. Emoticon features

An emoticon is a meta-communicative pictorial representation of a facial expression that uses punctuation marks such as *commas*, *hyphens* and *parenthesis* to convey a person's feelings or mood. Emoticons have become a very popular way of expressing sentiments on social media and thus presence of certain emoticons can act as useful indicator of sentiment in a Tweet. Motivated by (Mohammad et al., 2013), we develop a regular expression based on Christopher Potts tokenizing script [6] that can classify emoticons as having a positive sentiment or a negative sentiment.

Examples of some emoticons that can be recognized and classified by this script are given in Table 2.

For each Tweet in the dataset, we compute three binary-valued features that check

[4] http://www.umiacs.umd.edu/~saif/WebDocs/NRC-Hashtag-Sentiment-Lexicon-v0.1.zip

[5] http://www.umiacs.umd.edu/~saif/WebDocs/Sentiment140-Lexicon-v0.1.zip

[6] http://sentiment.christopherpotts.net/tokenizing.html

Positive	Negative	Facial Features Represented
:-)	:-(	Eyes, nose and mouth
:=)	:=(	Eyes, nose and mouth
=D	=(	Eyes and mouth only
=]	=[	Eyes and mouth only

Table 2: Illustrative example: Emoticons

whether a positive or negative emoticon is present in the Tweet; and whether the last token is positive or negative emoticon.

5. Encoding features:

   Tweet is a very non-conventional way of interaction which usually is a very noisy form of communication containing a lot of typos, ad-hoc abbreviations, phonetic substitutions, customized abbreviations, elongated words etc. These features combined with the above mentioned features can give useful insights in determining the sentiments of the Tweets.

   We implement the following encoding features:

   - **All-Caps:** Words with all characters in upper case represents a strong influence over the sentiment being conveyed. For example: "*You are a BIG loser!*" Here *BIG* puts extra emphasis on the person being called loser. We compute the feature value for each Tweet as the total number of words with all characters in upper case.
   - **Hashtags:** Tweets with hashtags contain additional information about the sentiment of the message being conveyed. For example: "*Twelve years a slave! #freedom #mustwatch*". This Tweet indirectly praises the movie named twelve years a slave for taking up the topic on slavery through hashtags. We compute the value of this feature to be equal to the total number of hashtags present in the Tweet.
   - **Elongated Words:** Presence of repetitive characters in a word strengthens the sentiment. For example: "*wohoooooo! I won!*", *"Ice-cream is so yummmmmy!"* contains elongated words signifying the extra emotion attached to the message. We compute the feature value as the total number of elongated words in each Tweet.
   - **Punctuations:** Tweets often contain the exclamation and question marks denoting sentiments according to their presence. For example: "*What happened to angels?? and demons?? and tigers and lions and beers ??!!!??!*". We define four features based on (i). number of contiguous sequences of exclamation marks; (ii). number of contiguous sequences of question marks; (iii). number of contiguous sequences of exclamation and question marks; and (iv). whether the last token in the Tweet is an exclamation or question mark.

2.3 Active Learning Framework

Active Learning can drastically reduce the amount of manual annotation by carefully selecting the most informative instances to be added to the initial training data. The key hypothesis behind active learning is that if we allow the learning algorithm to pick the data points that it finds very effective to learn, better results can be obtained on the task at hand. Therefore, these data should be carefully selected.

In the traditional random sampling approach, unlabelled data is selected for annotation at random. In a typical active learning setup, a classifier is trained on a small sample of data (usually selected randomly), known as the seed examples. The classifier is subsequently applied to a pool of unlabelled data with the purpose of selecting additional examples that the classifier views as informative. The selected data is annotated and the cycle is repeated, allowing the learner to quickly refine the decision boundary between the classes.

The main crux of the overall process is to determine the most informative samples. In our present work, we use the selection technique, which is based on the concept of *uncertainty sampling* which dictates that only those Tweets be added to the training set for which the classifier is most 'uncertain' about its classification. Selecting only such Tweets allow the classifier to iteratively learn from the informative samples, which are expected to yield an improved performance. In our framework we divide the datasets into three sets, *viz.* training, development and test. Classifier is

1: Evaluate the system on the gold standard test set.

2: Run SVM on the development set and compute the confidence score for each class for a target tweet.

3: For each tweet, compute the confidence interval as the difference between the scores of two most probable classes.

4: Sort the Tweets in ascending order of the confidence scores computed in Step 3.

5: Using predefined threshold number of the uncertain examples, select the Tweets from the development set to be added to the training set.

6: Remove the selected Tweets from the development set and add to the training set.

7: Re-train the SVM classifier with new training set and evaluate on the test set.

8: Repeat steps 2-7 until the performance in three consecutive iterations do not improve.

Figure 1: Active Learning Process

trained on the training set, and informative samples from the development set are chosen in such a way that the performance of the classifier improves on the test data when these are added to the initial training set. For each token of the development set, a SVM classifier produces the distance from the separating hyper-planes. Here at first we normalize these distance values in the range $[0, 1]$. The normalized value is treated as the confidence value for a particular class. Our uncertainty sample selection criteria is based on the concept of *margin sampling*. This is defined by the difference in the confidence scores of the two most probable classes predicted by the classifier, the hypothesis being that the Tweets for which this difference is little are the ones for which the classifier is less certain. We stop the process of active learning when the performance in three consecutive iterations do not improve.

The algorithm for active learning method is given in Figure 1.

3 Datasets, Experimental Results and Analysis

In this section, we describe the datasets used for the experiment, the results obtained for sentiment classification with different set-ups, and present the necessary analysis.

Sl. No.	Dataset	Positive Tweets	Negative Tweets	Total
1	Training Set	1,000	1,000	2,000
2	Develo-pment Set	1,000	1,000	2,000
3	Testing Set	1,000	1,000	2,000

Table 3: Details of base datasets used.

3.1 Description of Datasets

Our experiments are based on the datasets, obtained from *Twitter Sentiment Analysis Dataset* [7]. It is based on data from the following two sources: *University of Michigan Sentiment Analysis Competition on Kaggle* and *Twitter Sentiment Corpus* by Niek Sanders.

It contains 1,578,627 Tweets labelled with either positive (1) or negative (0). For our experiments we divide the dataset into training, development and test as shown in Table 3.

The details of the datasets mentioned above pertain to the first iteration of the active learning experiment. After the first iteration, the size of the training set varies based on the selection criteria while test set remains the same. The size of development set also varies accordingly.

3.2 Experiments and Results

The dataset is pre-processed to normalize it by converting *http://someurl* and all user *IDs* to *@someuser*. A baseline model is then developed by training SVM classifier on a balanced dataset of 2000 Tweets. Evaluation on a balanced test set (2000 tweets: 1000 positive and 1000 negative) shows the classification accuracy of 77.2%. Classification accuracies of positive and negative classes were 76.2% and 77.8%, respectively.

We perform active learning experiments varying the different parameters such as follows:

Experiment-1: We varied the threshold value denoting the number of least confident Tweets to be added to the training set for active learning.

Experiment-2: We attempted to estimate the optimum number of iterations required to attain a stable performance level.

[7] http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

Experiment-3: We studied the effect on the test set accuracy by varying the size of the development set.

Experiment-4: In this experiment, we compared our proposed method with a random selection approach where the uncertain samples are selected at random from the unlabelled data.

3.2.1 Experiment by varying the number of uncertain examples

Here we try to determine the optimum number of least confident Tweets which should be augmented to the training set in each iteration so that the optimum level of accuracy is achieved for the given test set. We iterate the algorithm by considering the various thresholds of the number of uncertain samples, i.e. 20, 50, 70, 100, 150, 225, 300, 400, 500 and 600. We observe the best accuracy with the threshold value of 100, i.e. in each iteration we add 100 most uncertain instances to the training data. This yields an overall accuracy of 83.95%.

In Figure 2 we plot how the accuracies and number of iterations are related with respect to the different threshold values. The figure demonstrates the results only up to the threshold value of 150. However, the nature of performance was quite similar for the other higher numbers too. We observe that rate of learning is higher when the threshold is set to larger value, denoting more number of uncertain Tweets are added.

Another interesting fact that can be observed from the graph is that for each of the cases, the accuracy improves steeply to a certain level (which is almost same for all, and this is around 83.5%) and then varies within the range of $\pm 0.5\%$ of the highest attained accuracy which is 83.95%. However, the number of iterations for higher number of augmentation of the most informative tweets to achieve that level of accuracy is less as evident from the graph. The algorithm executes for more iterations when the threshold values are lowered (for e.g., 20 or 50).

3.2.2 Experiments to estimate the optimum number of iterations and impact of data size

In this set-up of experiments we tried to estimate the optimum number of iterations that algorithm takes to achieve an optimum (near optimum) level of accuracy. We define optimum level of accuracy to be the level beyond which it does not improve even with more iterations. In particular we execute the algorithm for four consecutive iterations and stop if we observe no improvement in the accuracy or the improvemnt is within the range of $\pm 0.5\%$. Depending on the thresholds, the number of iterations could vary. In order to study the behaviour of the algorithm more extensively, we conducted the experiments by increasing the size of the development set to 5,000 tweets (2,500 positive and 2,5000 negative).

Iteration No.	DataSet I Accuracy	Dataset II Accuracy
1	77.20	77.20
2	79.30	78.05
3	80.20	78.95
4	81.25	79.90
5	81.50	80.10
6	82.20	80.15
7	82.65	80.90
8	83.05	81.30
9	82.95	81.45
10	83.40	81.42
11	83.80	81.30
12	83.85	81.40
13	83.95	81.20
14	83.75	81.55
15	83.80	81.48
16	83.80	81.60
17	83.80	81.65

Table 4: Results using two different samples. One, with training, test and development set each with 2,000 tweets (Dataset I) and the other with same training and test set but development set with 5,000 tweets (Dataset II)

Results are reported in Table 4 with the threshold set to 100, i.e. in each iterations 100 most uncertain instances are being added to the training set. The experimental results show that the number of iterations even for the same threshold of uncertain instances vary for two different development sets. It also appears that with the greater size in the development set, more number of iterations are required to reach the optimum level accuracy.

In order to estimate the impact of the dataset size we run the algorithm for 30 iterations. Results of first 16 iterations are given in Table 4, and the graphical representation for dataset II is plotted in Figure 3.

We observe that even with the increased data set, the accuracy remains between 81%-82%,

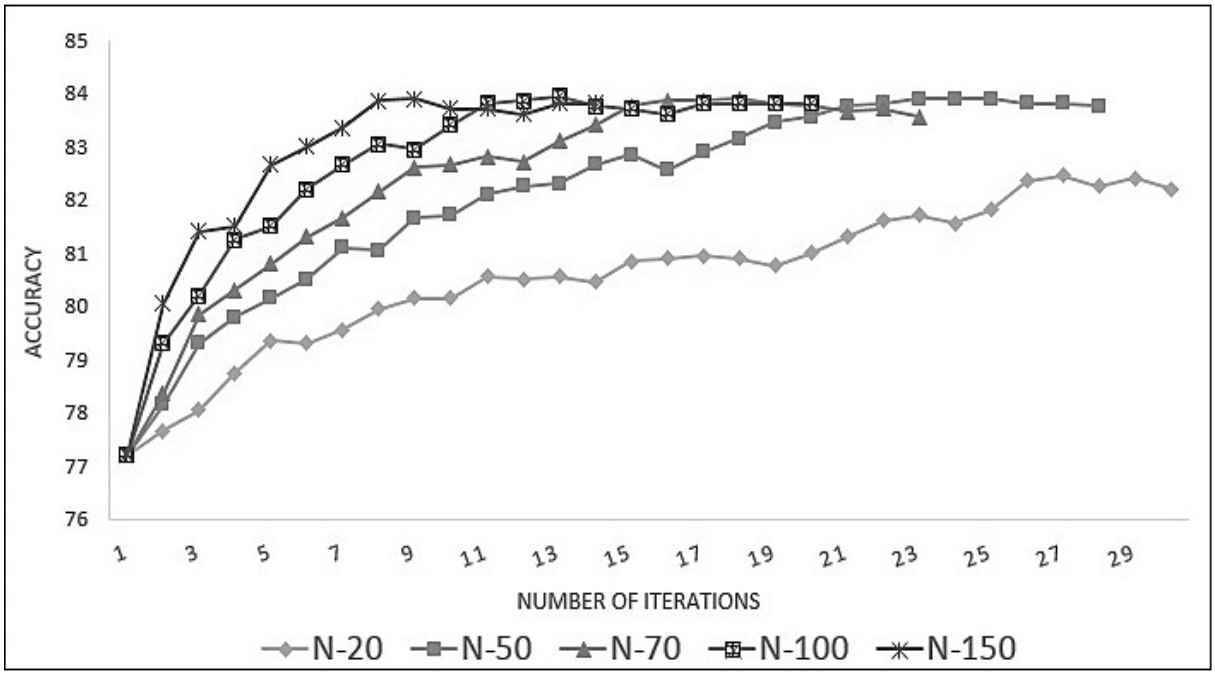

Figure 2: Graph of accuracy versus number of iterations for different cases of number of least confident Tweets to be added in the training set for active learning method.

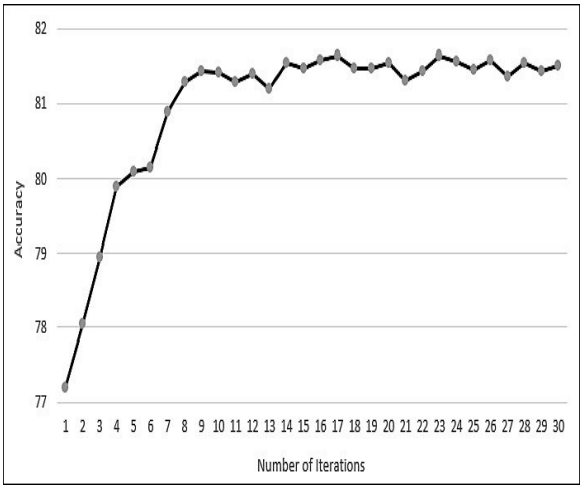

Figure 3: Graph representing the test set accuracy versus number of iterations for the dataset with development set containing 5000 samples.

while the highest test set accuracy obtained with the smaller dataset is 83.95%. This shows that the performance improvement of the target test set doesn't merely depend on the size of the unlabeled dataset but in the number of the informative instances selected to be added to the initial training data. Therefore unlabeled data should be selected in such a way that it relates to the target problem.

3.2.3 Comparing random selection method and active learning method

In this experimental setup we compare between the random sample selection and active learning methods. For active learning we pick up the best set-up, i.e, most uncertain 100 instances are augmented to the training set in each iteration. In case of random sampling approach, 100 instances are selected at random and added to the training set. From the results, reported in Table 5, it is evident that active learning technique is more effective compared to the random selection technique.

3.3 Comparison with related works

Active Learning based sentiment analysis is a recent topic of research and a few methods has been proposed for different datasets and varying contexts. The authors (Li et al., 2012) discussed how active learning can be used to handle the imbalanced dataset containing Amazon product reviews. The active learning methodology they adopt have two classifiers for selecting most informative minority samples. They use various active learning strategies based on random, margin, uncertainty, certainty, co-testing and self-selecting selections to improve the accuracy of the imbalanced dataset and were able to correctly annotate more than 90% of the samples using co-selection strategy. Another method was proposed in (Li et

Itn.	Training Set	Accuracy in AL Method	Accuracy in RS Method
1	2000	77.20	77.20
2	2100	79.30	77.85
3	2200	80.20	78.75
4	2300	81.25	79.70
5	2400	81.50	80.70
6	2500	82.20	80.60
7	2600	82.65	81.00
8	2700	83.05	81.15
9	2800	82.95	81.36
10	2900	83.40	81.79
11	3000	83.80	81.93
12	3100	83.85	82.20
13	3200	83.95	82.34
14	3300	83.75	82.67

Table 5: Results comparing Active Learning (AL) method and Random Selection (RS) method.

al., 2013) that makes use of the same dataset, and studies the active learning for cross-domain sentiment classification. The classifiers employ the selection strategy of Query By Committee (QBC) and use label propagation (LP) algorithm for training.

However, there has not been a significant study on the active learning based document level sentiment classification of Tweets. Also we can't directly compare the accuracies obtained in the works (Li et al., 2012; Li et al., 2013) with the ones we obtained as the experimental setups are different.

4 Conclusion and Future Work

In this paper, we have investigated the effectiveness of active learning method for sentiment classification of Tweets. At first a SVM based sentiment classifier has been developed with the set of features useful for classifying Tweets into positive and negative sentiments. Our active learning algorithm is based on the criterion of margin sampling. We observe an overall performance improvement of 6.75% over the baseline model. Our results suggest, first of all, that active learning does lead to better results than random sampling; and second, that our approach leads to reasonable results with relatively small amount of useful data. We observed that the rate of performance improvement doesn't depend on the larger samples selected to be added to the training set, but on the amount of informative Tweet samples.

In future work, we plan to introduce some more features for sentiment classification. In present work we used uncertainty margin sampling as he selection criteria of most informative samples which yielded satisfactory results, however, we plan to employ other sampling techniques in future work and devise new stopping criteria for the algorithm.

References

Satanjeev Banerjee and Ted Pedersen. 2003. The design, implementation, and use of the ngram statistics package. In *Computational Linguistics and Intelligent Text Processing*, pages 370–381. Springer.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Cynthia Chew and Gunther Eysenbach. 2010. Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak. *PloS one*, 5(11):e14118.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Asif Ekbal and Sriparna Saha. 2012. Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(2):143–166.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541.

Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Li. 2012. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148. Association for Computational Linguistics.

Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2127–2133. AAAI Press.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Saif M Mohammad. 2012. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390.

Alexander Pak and Patrick Paroubek. 2010. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 436–439. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bernhard Schölkopf, Christopher JC Burges, and Alexander J Smola. 1999. *Advances in kernel methods: support vector learning*. MIT press.

Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66.

Sudha Verma, Sarah Vieweg, William J Corvey, Leysia Palen, James H Martin, Martha Palmer, Aaron Schram, and Kenneth Mark Anderson. 2011. Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In *ICWSM*.

Jason Weston, Chris Watkins, et al. 1999. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224.