IITPSemEval: Sentiment Discovery from 140 Characters

Ayush Kumar, Vamsi Krishna Akella and Asif Ekbal
Department of Computer Science and Engineering
Indian Institute of Technology, Patna
Patna, 800013
Email: ayush.cs12@iitp.ac.in, vamsi.cs11@iitp.ac.in, asif@iitp.ac.in

Abstract

This paper presents an overview of the system developed and submitted as a part of our participation to the SemEval-2015 Task 10 that deals with Sentiment Analysis in Twitter. We build a Support Vector Machine (SVM) based supervised learning model for Subtask A (term level task) and Subtask B (message level task). We also participate in Subtask E viz., determining degree of polarity, and build a very simple system by employing the available lexical resources. Experiments with the 2015 official datasets show F1 scores of 81.31% and 58.80% for Task A and Task B, respectively. For Subtask E, our model achieves a score of 0.413 on Kendal's Tau metric.

1 Introduction

The use of social media platforms has become central to many teenager's and adult's lives. With the emerging forms of communication, much of the freely available texts in the opinionated texts are linguistically unstructured. People have adopted creative spellings and abbreviations, and are excessively using more intelligent forms of messages that involves typos, hash-tags and emoticons to convey their messages. The huge abundance of inexpensive data, rich in applications, can prove handy for public and corporate institutions. This has urged the scientific community to extract the substantive information from these texts. The proliferation of microblogging sites like Twitter which boasts of user's comments on everything trending in real time opens up an unprecedented opportunity to explore and develop techniques to mine the information.

Task 10 in Semantic Evaluation 2015 provides a research platform promoting the knowledge discovery in Twitter. Task 10 consists of five different subtasks: Contextual Polarity Disambiguation (A), Message Polarity Classification (B), Topic-Based Message Polarity Classification (C), Detecting Trends Towards a Topic (D) and Determining degree of polarity of Twitter terms with the sentiment (E). Complete details of the task can be found at (Rosenthal et al., 2015). We participated in Subtasks A, B and E, the first two of which require the sentiments to be classified into positive, negative and neutral classes for a given segment of the tweet (for A) or the entire message (for B), while the Task E needs to compute the strength of association of the given terms to the sentiment on a scale of 0 to 1 with 1 denoting the maximum strength.

The technical study of public sentiment has been a subject of trending research and a significant amount of extensive work is being carried out in the domain. Sentiment Analysis has been handled at the various levels of granularity. Early research works (Pang and Lee, 2004) focussed on the document level classification with further studies at message and term level (Rosenthal et al., 2014). Twitter has also been investigated for its possible applications in the fields of commerce (Jansen et al., 2009; Bollen et al., 2011), elections (O'Connor et al., 2010; Tumasjan et al., 2010), disaster management (Nagy and Stamberger, 2012; Terpstra et al., 2012) etc. using varied approaches and different experimental setups. Semantic Evaluation tasks (Nakov et al., 2013; Rosen-

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 601–607,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

thal et al., 2014) continue to pitch in with the newer systems for the sentiment classification of tweets.

2 Proposed Approach

In this section, we describe the supervised learning system that we develop for the first two subtasks, namely A and B. The first section would focus on Tasks A and B and later section would describe the method that was adopted for Task E.

2.1 Preprocessing

We normalize all URLs to http://someurl and all usernames to @someuser. We also pre-process the dataset to convert character encodings like \u2019(’), \u002c(,) &amp;(&), &lt;(<), &gt;(>), &nbsp;(whitespace), <3(love) etc. to their usual text so as to reduce the noise.

2.2 Methods for Contextual Disambiguation and Message Classification

We develop the methods for the first two tasks based on supervised Support Vector Machine (Cortes and Vapnik , 1995).

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: $(\mathbf{w}.\mathbf{x}) + b = 0 \quad \mathbf{w} \in \mathbf{R}^D, b \in \mathbf{R}$.

In our work we make use of the SVM implementation as available with the LibLinear [1] model (Fan et al., 2008). LibLinear has been optimized for data with millions of instances with very large feature spaces. To develop the feature-based learning model, we categorize the features into three groups: Token-level Features (Group-I), Semantic Features (Group-II) and Encoding Features (Group-III).

The set of features that we implement for the target tasks are described as follows.

1. **Group-1: Token-level Features**: These correspond to the features like n-grams and Part-of-Speech (PoS).

    - **Word n-grams**: All n-grams of sizes 1 and 2 are extracted for Task A using Ngram Statistics Package (Banerjee and Pedersen, 2003). This binary valued feature is implemented as contextual feature for Task A. Based on the results obtained on the development set, two words on each side of the targeted segment are taken into consideration. For Task B, all n-grams of size upto three are extracted.
    - **Character n-Grams**: For each token in the target text in the tweet, all the character n-grams of prefix and suffix of lengths of two and three characters are extracted. This feature is implemented only for the term level task.
    - **Part of Speech (PoS) Information**: For both the subtasks, we label each token in the tweet with CMU ARK PoS tagger (Gimpel et al., 2011). The number of each of the PoS tags is kept as feature.

2. **Group-II: Semantic Features**: To take into account the semantics of the text present in the tweet/targeted segment, we use Lexicon and SentiWordNet based features.

    - **Lexicon Features**: We use lexicons such as NRC Hashtag [2], Sentiment 140 [3], Bing Liu (Liu et al., 2005) and NRC Emotion Lexicons (Mohammad and Turney, 2013) to implement various features. The implementation of features for these tasks is based on the number of tokens associated with positive and negative sentiment using NRC Hashtag, Sentiment 140 and Bing Liu lexicon. For NRC Hashtag and Sentiment 140 lexicon, the sentiment scores of the tokens are used to implement total score of the message as another feature.

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

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

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

The NRC emotion lexicon is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy and disgust) and two sentiments (negative and positive). We categorize joy, surprise, trust and anticipation as positive emotions and the rest as negative emotions. Based on the categorization, we compute the number of tokens with positive score, number of tokens with negative score and number of tokens with neutral score as the features.

- **SentiWordNet Feature**: We compute the average positive score (posScore) and negative score (negScore) for each word in the tweet using SentiWordNet3.0 (Baccianella et al., 2010). For a given tweet we define two features that denote the number of words which have posScore greater than negScore, and number of word with negScore greater than posScore.
- **Inverted Segment**: An inverted segment is defined as the part of the tweet which occurs after an inverting word (i.e. the tokens that denote the negative context) such as doesn't, isn't, can't, etc. until a punctuation. The polarity of the words occurring in the inverted segment is reversed, i.e. a token with positive or negative sentiment is converted to the token bearing negative or positive sentiment, respectively. The intensity values of the tokens are adopted from the NRC Hashtag lexicon (Mohammad et al., 2013) and Sentiment140 lexicon (Mohammad et al., 2013) which are used to construct the feature vector. The feature vector contains several pieces of information that denote the number of inverted segments in the tweet, sum of intensities of all the words that appear in the inverted segments in the tweet, etc.
- **Tweet Clusters**: We use the CMU Twitter Word Clusters (Owoputi et al., 2013) to generate the clusters of words that appear either in the context of positive or negative sentiment. All the tokens which belong to the positive sub-cluster occur more in positive context than in negative context. Similarly all the tokens which belong to the negative sub-cluster occur more in negative context than in the positive context. The categorization of positive and negative sub-cluster is done based on the number of times the token occurs in positive and negative contexts. A feature vector of length 2000 is defined, each bit of which takes a value denoting the number of times the token appears in the tweet.

3. **Group-III: Encoding Features**: The text of the tweet is normally different from the general English text. It contains emoticons, hashtags, repetitive characters and irregular punctuations. To incorporate these encodings, we implement the following features.

   - **Intensifiers**: There are several words that denote the intensity of sentiment, and these can be used as the features of the model. We use the number of hashtags, number of words in uppercase (e.g. BIG loser) and number of elongated words (e.g. yummmmmy) in the tweet as the features. These features were used for both the tasks.
   - **Emoticon Features**: This is a binary valued feature that denotes the presence or absence of the positive and negative emoticon.
   - **Punctuation**: The number of occurrences of contiguous sequences of question marks (????), exclamation marks (!!!) and question-exclamation marks (?!!?) are extracted from the tweet. This feature is not used for subtask A as we observe lower performance of the system on the development set.
   - **URL and Username**: This feature takes into account the number of occurrences of the username and URLs. The feature is defined for the term level task.

2.3 Method for Determining the Strength

Our approach for determining the strength of sentiment bearing words is based on the rule-based ap-

Set	Positive	Negative	Neutral
Training	5480	2967	434+434
Development	648	430	57
2015 Test	1896	1006	190
Progress Test	6354	3771	556

Table 1: Dataset for Task A.

Set	Positive	Negative	Neutral
Training	3064	1204	3942
Development	575	340	739
2015 Test	1038	365	987
Progress Test	3506	1541	3940

Table 2: Dataset for Task B.

proach that is developed using various available resources. We use the sentiment scores of terms extracted from SentiWordNet, Sentiment140 bigram lexicon and NRC Hashtag unigram lexicon. In these lexicons, terms have been assigned scores based on their association to the positive or negative sentiment in some contexts. We also observe that out of the 200 words present in the trial data, 167 words are present at least in one of these three lexicons, which is more than 83%. This is why we use these resources for subtask E.

At first we extract the scores of the given term from the SentiWordNet. The scores denote the associativity of the word towards the positive and negative sentiment in various contexts. Let us assume that posScore and negScore denote the positive and negative scores of the target word, respectively. We compute the average positive and negative scores of all the terms, and the final score is set as Score = Avg posScore - Avg negScore.

If the word or term is not available in the SentiWordNet, we look at the Sentiment140 or NRC Hashtag lexicon. The score of each term in these lexicons corresponds to the number of times the term co-occurs with the positive and negative sentiment. For unigram we search in the NRC Hashtag lexicon, and for the others we look at Sentiment140 lexicon. The score of each term is set as: Score = (no. of positive occurrences - no. of negative occurrences)/(no. of positive occurrences + no. of negative occurrences). For the word that does not appear in any of these lexicons, we assign the default score of 0.5. If the range of the scores is between -1 to 1, we normalize the values between 0 and 1.

3 Datasets and Experimental Results

To train and tune our system, we use the training and development datasets that were employed for Task 2 in SemEval 2013 (Nakov et al., 2013). The system is tested on two datasets for this year's tasks, one is the progress set and the other one is the 2015 official test set. The datasets are annotated with three classes, namely positive, negative and neutral. The training sets consist of 9,315 and 8,210 annotated tweets for subtask A and B, respectively. The progress set contains tweets from five different categories: LiveJournal 2014, SMS 2013, Twitter 2013, Twitter 2014 and Twitter 2014 Sarcasm. The datasets used for the Tasks A and B are summarized in Table 1 and Table 2, respectively. The metric used for evaluating the system is average F1-score (averaged F1-positive and averaged F1-negative, and ignoring the F1-neutral) for 2015 test set, while the ranking for progress set is done on the F1 score of the Twitter 2014 subset.

For Task E, the trial dataset comprise of 200 unique words/phrases with the corresponding scores denoting the strength of the terms with positive or negative sentiment. The test set contains 1,315 words/phrases which has to be scored in between 0 to 1 indicating their association with the positive or negative sentiment.

We observe that proportion of neutral tweets in the training set of Task A is quite less (4.88%). In order to create a balanced dataset, we perform oversampling to increase the number of neutral tweets in the training data. Experiments are carried out with various oversampling rates. Based on the evaluation on the development data, we observe that oversampling the neutral tweets by increasing its number twice lead to better scores while constructing the dataset with thrice the number of neutral tweets results in over-fitting, and hence, lowers the F1-score value. For the second task, we also perform this oversampling technique for the better representations of negative tweet instances. However we notice a reduction in the overall F1-score compared to the performance that we achieved with our original

Features	F1-Score: Task A	F1-Score: Task B
All	81.31	58.80
All-Token	80.04 (-1.27)	54.51 (-4.29)
All-Semantic	76.09 (-5.22)	48.29 (-10.51)
All-Encoding	81.18 (-0.13)	58.24 (-0.56)
All-WordNgram	80.75 (-0.56)	54.92 (-3.88)
All-CharNgram	81.25 (-0.06)	-
All-Ngram	80.30 (-1.01)	54.92 (-3.88)
All-POS	81.23 (-0.08)	59.10 (+0.30)
All-NRCHashtag	81.23 (-0.08)	57.31 (-1.49)
All-Senti140	81.93 (+0.62)	56.73 (-2.07)
All-Bing	80.91 (-0.40)	56.16 (-2.64)
All-Emotion	81.01 (-0.30)	57.68 (-1.12)
All-Lexicon	80.19 (-1.12)	43.23 (-15.57)
All-Cluster	81.24 (0.07)	55.62 (-3.18)
All-Inverted	81.37 (+0.06)	58.73 (-0.07)
All-SentiWord	81.14 (-0.17)	58.44 (-0.36)
All-Intensifiers	81.22 (-0.09)	58.49 (-0.31)
All-Emoticon	81.25 (-0.06)	58.33 (-0.47)
All-URL/Username	81.31 (0.0)	-
All-Punctuation	-	58.64 (-0.16)

Table 3: Experimental results for feature-ablation experiment for Task A and B. The values in the parenthesis denotes the deviation from the score when all the features were taken into consideration.

setup.

For subtask A, our system achieves a F1-score of 81.31% for 2015 test set and 82.73% for Twitter 2014 subset of progress set. For the message level task, i.e. for subtask B, our system obtains the F1-scores of 58.80% for the 2015 test set and 65.09% for the progress test set. The best ranked team for the term level task shows the F1-score of 84.79% for the 2015 test set and 87.12% for the progress test. For Subtask B, the best performing system produces the F1-scores of 64.84% for the 2015 test set and 74.42% for the progress set.

For Task E, we have to provide a score between 0 and 1 for a word or phrase denoting the associativity of the phrase with the positive sentiment. The evaluation metric used for this task is based on Kendall's Tau rank correlation coefficient. Our model obtains a score of 0.413 with respect to the best team's score of 0.625.

3.1 Feature Engineering and Analysis of Results

We observe that our system performs much better for the term level task than the message level task. This can be contributed to the fact that the contextual polarity disambiguation is, in general, single sentiment oriented whereas a message level sentiment classification is ambiguous because of the tweet containing mixed sentiments. To get an insight to the contribution of each feature in development of the system, we perform feature engineering. Experiments of the detailed feature ablation study are shown in Table 3.

From the feature ablation experiment, we observe that in both the tasks, semantic features (i.e. sentiment lexicons) contribute significantly. Among semantic features, both Task A and B rely heavily on lexicon features. It can also be noted that the encoding features which are characteristics of twitter text also help in marginal improvement.

However, the inverted segment feature does not result in the expected performance gain. This can be explained in light of the following two aspects. Let us consider the two statements as: (a) *The coffee tastes bad.* and (b) *The book is not bad.*, the first statement signifies negative sentiment while the second statement is neutral. However, if we take into account the method that we adopted, in sentence (b) according to our approach a negative word (*bad*) becomes positive with the same intensity as we only invert the polarity without changing the intensity of the word, but in this sentence *bad* actually becomes neutral when it occurs in an inverted segment (i.e. after 'not'). Another reason might be the possible conflict between the lexicon and inverted segment features. In lexicon feature, we consider the scores of each token for generating the feature vector where the word '*bad*' is taken into negative sense for both the cases.

3.2 Conclusions and Future Work

In this paper we describe our systems that we developed as part of our participation to the SemEval shared task on Sentiment Analysis on Twitter. Out of the five defined tasks, we participated in three tasks. We have developed a supervised SVM model for the contextual polarity disambiguation (Task A) and message level sentiment classification (Task B). Our system showed promising results for the Task A and satisfactory performance for Task B. However, when we did feature ablation experiment, we found that certain features (like inverted segment) did not contribute substantially as expected. In our future work, we will try to address this issue. The n-grams feature that we have used, generates sparse feature vector. Proper smoothing techniques might be helpful to reduce the noise in the feature vector due to the sparsity in the n-grams feature. Apart from this, we also plan to develop a method in order to automatically identify the most relevant set of features for the individual tasks.

Our approach for the Task E was purely based on the rules that we derived from the various available resources. The lexicons that we used have different ranking schemes, i.e. the same term can have different ranks based on its sentiment intensity as present in the different lexicons. We are exploring to come up with the appropriate method to merge the different ranks obtained from the different lexicons. Some other resources like NRC Emotion lexicon and MPQA Subjectivity lexicon can also be used. Other future works include developing methods for tasks C and D.

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector Networks. Machine Learning. Volume 20, pages. 273–297.

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. HLT-NAACL. pages. 380–390.

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. LREC. pages. 2200–2204.

Bing Liu, Minqing Hu and Junsheng Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th International World Wide Web conference*. Journal of Machine Learning Research. May 10-14. Chiba, Japan

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. Book-title: Computational Intelligence. Volume 29, pages. 436–465.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A Library for Large Linear Classification. Journal of Machine Learning Research. Volume 9, pages. 1871-1874.

Sara Rosenthal, Alan Ritter, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation* Denver, Colorado, June 2015.

Sara Rosenthal, Alan Ritter, Preslav Nakov and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pages 73–80, Dublin, Ireland, August 2014.

Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov and Theresa Wilson. 2013. Semeval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. pages 312–320, Atlanta, Georgia, USA, June 2013.

Bernard J Jansen, Mimi Zhang, Kate Sobel and Abdur Chowdury. 2009. Twitter Power: Tweets as Electronic Word of Mouth. In *Journal of the American Society for Information Science and Technology*. Volume 60, Number 11, pages 2169–2188.

Johan Bollen, Huina Mao and Xiaojun Zeng. 2011. Twitter Mood Predicts The Stock Market. In *Journal of Computational Science*. Volume 10, Number 1 pages 1–8.

Teun Terpstra, A de Vries, R Stronkman and GL Paradies. 2012. Towards a Realtime Twitter Analysis during Crises for Operational Crisis Management. In *Proceedings of the 9th International ISCRAM Conference*.

Kevin Gimpel, Nathan Schneider, Brendan OConnor, 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*. pages 42–47.

Satanjeev Banerjee and Ted Pedersen. 2003. The Design, Implementation, and Use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*. pages 370–381, February 2003, Mexico City.

Ahmed Nagy and Jeannie Stamberger. 2012. Crowd Sentiment Detection During Disasters and Crises. In *Proceedings of the 9th International ISCRAM Conference*. pages 1–9.

Saif M Mohammad, Svetlana Kiritchenko and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises*. 2013.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *ICWSM*. Volume 10, pages 178–185.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. page 271.

Brendan O'Connor, Ramnath Balasubramanyanand, Bryan R Routledge and Noah A Smith. 2010. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *ICWSM*. Volume 11, pages 122–129.