IIIT-H at SemEval 2015: Twitter Sentiment Analysis The good, the bad and the neutral!

Ayushi Dalmia, Manish Gupta[*], Vasudeva Varma
Search and Information Extraction Lab
International Institute of Information Technology, Hyderabad
{ayushi.dalmia@research.iiit.ac.in, manish.gupta@iiit.ac.in, vv@iiit.ac.in }

Abstract

This paper describes the system that was submitted to SemEval2015 Task 10: Sentiment Analysis in Twitter. We participated in Subtask B: Message Polarity Classification. The task is a message level classification of tweets into positive, negative and neutral sentiments. Our model is primarily a supervised one which consists of well designed features fed into an SVM classifier. In previous runs of this task, it was found that lexicons played an important role in determining the sentiment of a tweet. We use existing lexicons to extract lexicon specific features. The lexicon based features are further augmented by tweet specific features. We also improve our system by using acronym and emoticon dictionaries. The proposed system achieves an F1 score of 59.83 and 67.04 on the Test Data and Progress Data respectively. This placed us at the 18^{th} position for the Test Dataset and the 16^{th} position for the Progress Test Dataset.

1 Introduction

Micro-blogging has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life, everyday on popular websites such as Twitter, Tumblr and Facebook. Spurred by this growth, companies and media organizations are increasingly seeking ways to mine these social media for information about what people think about their companies and products. Political parties may be interested to know if people support their program or not. Social organizations may need to know people's opinion on current debates. All this information can be obtained from micro-blogging services, as their users post their opinions on many aspects of their life regularly.

Twitter contains an enormous number of text posts and the rate of posts is increasing every day. Its audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interest groups. However, analyzing Twitter data comes with its own bag of difficulties. Tweets are small in length, thus ambiguous. The informal style of writing, a distinct usage of orthography, acronymization and a different set of elements like hashtags, user mentions demand a different approach to solve this problem.

In this work we present the description of the supervised machine learning system developed while participating in the shared task of message based sentiment analysis in SemEval 2015 (Rosenthal et al., 2015). The system takes as input a tweet message, pre-processes it, extracts features and finally classifies it as either positive, negative or neutral. Tweets in the positive and negative classes are subjective in nature. However, the neutral class consists of both subjective tweets which do not have any polarity as well as objective tweets.

Our paper is organized as follows. We discuss related work in Section 2. In Section 3, we discuss the existing resources which we use in our system. In Section 4 we present the proposed system and give a detailed description for the same. We present experimental results and the ranking of our system for different datasets in Section 5. The paper is summarized in Section 6.

2 Related Work

Sentiment analysis has been an active area of research since a long time. A number of surveys (Pang and Lee, 2008; Liu and Zhang, 2012) and books (Liu, 2010) give a thorough analysis of the existing techniques in sentiment analysis. Attempts have been made to analyze sentiments at different levels starting from document (Pang and Lee, 2004),

[*] The author is also a researcher at Microsoft (gmanish@microsoft.com)

sentences (Hu and Liu, 2004) to phrases (Wilson et al., 2009; Agarwal et al., 2009). However, micro-blogging data is different from regular text as it is extremely noisy in nature. A lot of interesting work has been done in order to identify sentiments from Twitter micro-blogging data also. (Go et al., 2009) used emoticons as noisy labels and distant supervision to classify tweets into positive or negative class. (Agarwal et al., 2011) introduced POS-specific prior polarity features along with using a tree kernel for tweet classification. Besides these two major papers, a lot of work from the previous runs of the SemEval is available (Rosenthal et al., 2014; Nakov et al., 2013).

3 Resources

3.1 Annotated Data

Tweet IDs labeled as positive, negative or neutral were given by the task organizers. In order to build the system we first downloaded these tweets. The task organizers provided us with a certain number of tweet IDs. However, it was not possible to retrieve the content of all the tweet IDs due to changes in the privacy settings. Some of the tweets were probably deleted or may not be public at the time of download. Thus we were not able to download the tweet content of all the tweets IDs provided by the organizers. For the training and the dev-test datasets, while the organizers provided us with 9684 and 1654 tweet IDs respectively, we were able to retrieve only 7966 and 1368 tweets, respectively.

3.2 Sentiment Lexicons

It has been found that lexicons play an important role in determining the polarity of a message. Several lexicons have been proposed in the past which are used popularly in the field of sentiment analysis. We use the following lexicons to generate our lexicon based features: (1) Bing Liu's Opinion Lexicon[1], (2) MPQA Subjectivity Lexicon (Wilson et al., 2005), (3) NRC Hashtag Sentiment Lexicon (Mohammad et al., 2013), and (4) Sentiment140 Lexicon (Mohammad et al., 2013).

3.3 Dictionary

Besides the above sentiment lexicons, we used two other dictionaries described as follows.

- **Emoticon Dictionary:** We use the emoticons list [2] and manually annotate the related sentiment. We categorize the emoticons into four classes as follows: (1) Extremely- Positive, (2) Positive, (3) Extremely- Negative, and (4) Negative.

- **Acronym Dictionary:** We crawl the noslang.com website [3] in order to obtain the acronym expansion of the most commonly used acronyms on the web. The acronym dictionary helps in expanding the tweet text and thereby improves the overall sentiment score. The acronym dictionary has 5297 entries. For example, *asap* has the translation *As soon as possible*.

Other than this we also use Tweet NLP (Owoputi et al., 2013), a Twitter specific tweet tokenizer and tagger which provides a fast and robust Java-based tokenizer and part-of-speech tagger for Twitter.

4 System Overview

Figure 1 gives a brief overview of our system. In the offline stage, the system takes the tweet IDs and the N-Gram model as inputs (shown in red) to learn a classifier. The classifier is then used online to process a test tweet and output (shown in green) its sentiment. The basic building blocks of the system include Pre-processing, Feature Extraction and Classification. We first build a baseline model based on unigram, bigrams and trigrams and later add more features to it. In this section we discuss each module in detail.

4.1 Pre-processing

Since the tweets are very noisy, they need a lot of pre-processing. Table 1 lists the various steps of pre-processing applied on the tweets. They are discussed as follows.

- *Tokenization*
  After downloading the tweets using the tweet IDs provided in the dataset, we first tokenize them. This is done using the Tweet-NLP tool (Gimpel et al., 2011) developed by ARK Social Media Search. This tool tokenizes the

[1] `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`

[2] `http://en.wikipedia.org/wiki/List_of_emoticons`

[3] `http://www.noslang.com/dictionary/`

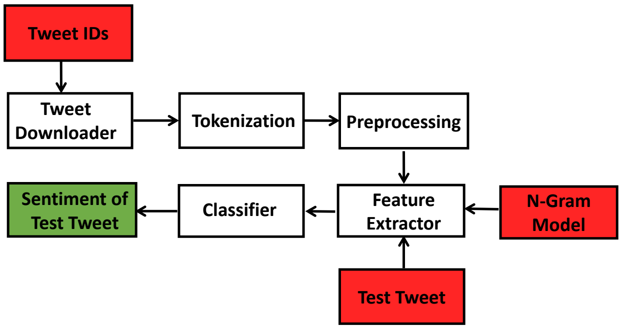

Figure 1: System Architecture (Red: Inputs, Green: Outputs).

Table 1: List of Pre-processing Steps.

Tokenisation
Remove Non-English Tweets
Replace Emoticons
Remove Urls
Remove Target Mentions
Remove Punctuations from Hashtags
Handle Sequences of Repeated Characters
Remove Numbers
Remove Nouns and Prepositions
Remove Stop Words
Handle Negative Mentions
Expand Acronyms

tweet and returns the POS tags of the tweet along with the confidence score. It is important to note that this is a Twitter specific tagger and tags the Twitter specific entries like emoticons, hashtags and mentions along with the regular parts of speech. After obtaining the tokenized and tagged tweets, we move to the next step of preprocessing.

- *Remove Non-English Tweets*
  Twitter allows more than 60 languages. However, this work currently focuses on English tokens only. We remove the tweets with non-English tokens.

- *Replace Emoticons*
  Emoticons play an important role in determining the sentiment of the tweet. Hence we replace the emoticons by their sentiment polarity by looking up in the Emoticon Dictionary generated using the dictionary mentioned in Section 3.

- *Remove Urls*
  The urls which are present in the tweet are shortened due to the limitation on the length of the tweet text. These shortened urls do not carry much information regarding the sentiment of the tweet. Thus these are removed.

- *Remove Target Mentions*
  The target mentions in a tweet done using '@' are usually the twitter handle of people or organizations. This information is also not needed to determine the sentiment of the tweet. Hence they are removed.

- *Remove Punctuations from Hashtags*
  Hashtags represent a concise summary of the tweet, and hence are very critical. In order to capture the relevant information from hashtags, all special characters and punctuations are removed before using them as a feature.

- *Handle Sequences of Repeated Characters*
  Twitter provides a platform for users to express their opinion in an informal way. Tweets are written in a noisy form, without any focus on correct structure and spelling. Spell correction is an important part in sentiment analysis of user-generated content. People use words like 'coooool' and 'hunnnnngry' in order to emphasize the emotion. In order to capture such expressions, we replace the sequence of more than three similar characters by three characters. For example, 'wooooow' is replaced by 'wooow'. We replace by three characters so as to distinguish words like 'wow' and 'wooooow'.

- *Remove Numbers*
  Numbers are of no use when measuring sentiment. Thus, numbers which are obtained as tokenized units from the tokenizer are removed in order to refine the tweet content.

- *Remove Nouns and Prepositions*
  Given a tweet token, we identify the word as a noun word by looking at its part-of-speech

tag assigned by the tokenizer. If the majority sense (most commonly used sense) of that word is noun, we discard the word. Noun words do not carry sentiment and thus are of no use in our experiment. Similarly we remove prepositions too.

- *Remove Stop Words*
  Stop words play a negative role in the task of sentiment classification. Stop words occur in both positive and negative training set, thus adding more ambiguity in the model formation. Also, stop words do not carry any sentiment information and thus are of no use.

- *Handle Negative Mentions*
  Negation plays a very important role in determining the sentiment of the tweet. Tweets consist of various notions of negation. Words which are either 'no', 'not' or ending with 'n't' are replaced by a common word indicating negation.

- *Expand Acronyms*
  As described in Section 3 we use an acronym expansion list. In the pre-processing step we expand the acronyms if they are present in the tweet.

4.2 Baseline Model

We first generate a baseline model as discussed in (Bakliwal et al., 2012). We perform the pre-processing steps listed in Section 4.1 and learn the positive, negative and neutral frequencies of unigrams, bigrams and trigrams in our training data. Every token is given three probability scores: Positive Probability (P_p), Negative Probability (N_p) and Neutral Probability (NE_p). Given a token, let P_f denote the frequency in positive training set, N_f denote the frequency in negative training set and NE_f denote the frequency in neutral training set. The probability scores are then computed as follows.

$$P_p = \frac{P_f}{P_f + N_f + NE_f} \tag{1}$$

$$N_p = \frac{N_f}{P_f + N_f + NE_f} \tag{2}$$

$$NE_p = \frac{NE_f}{P_f + N_f + NE_f} \tag{3}$$

Next we create a feature vector of tokens which can distinguish the sentiment of the tweet with high confidence. For example, presence of tokens like *am happy!*, *love love* , *bullsh*t !* helps in determining that the tweet carries positive, negative or neutral sentiment with high confidence. We call such words, **Emotion Determiner**. A token is considered to be an Emotion Determiner if the probability of the emotion for any one sentiment is greater than or equal to the probability of the other two sentiments by a certain threshold. It is found that we need different thresholds for unigrams, bigrams and trigrams. The threshold parameters are tuned and the optimal threshold values are found to be 0.7, 0.8 and 0.9 for the unigram, bigram and trigram tokens, respectively. Note that before calculating the probability values, we filter out those tokens which are infrequent (appear in less than 10 tweets). This serves as a baseline model. Thus, our baseline model is learned using a training dataset which contains for every given tweet, a binary vector of length equal to the set of Emotion Determiners with 1 indicating its presence and 0 indicating its absence in the tweet. After building this model we will append the features discussed in Section 4.3. After appending the features to the baseline model, we get enhanced richer vectors containing Emotion Determiners along with the new feature values.

4.3 Feature Extraction

We propose a set of features listed in Table 2 for our experiments. There are a total of 34 features. We calculate these features for the whole tweet in case of message based sentiment analysis. We can divide the features into two classes: a) Tweet Based Features, and b) Lexicon Based Features. Table 2 summarizes the features used in our experiment. Here features $f_1 - f_{22}$ are tweet based features while features $f_{23} - f_{34}$ are lexicon based features.

A number of our features are based on prior polarity score of the tweet. For obtaining the prior polarity of words, we use AFINN dictionary [4] and extend it using SENTIWORDNET (Esuli and Sebastiani, 2006). We first look up the tokens in the tweet in the AFINN lexicon. This dictionary of about 2490 English language words assigns every word a pleasantness score between -5 (Negative) and +5 (Posi-

[4] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

Table 2: Description of the Features used in the Model.

Feature Description	**Feature ID**
Prior Polarity Score of the Tweet	f_0
Brown Clusters	f_1
Percentage of Capitalised Words	f_2
# of Positive Capitalised Words	f_3
# of Negative Capitalised Words	f_4
Presence of Capitalised Words	f_5
# of Positive Hashtags	f_6
# of Negative Hashtags	f_7
# of Positive Emoticons	f_8
# of Extremely Positive Emoticons	f_9
# of Negative Emoticons	f_{10}
# of Extremely Negative Emoticons	f_{11}
# of Negation	f_{12}
# Positive POS Tags	f_{13}
# Negative POS Tags	f_{14}
Total POS Tags Score	f_{15}
# of special characters like ? ! and *	$f_{16}.f_{17}, f_{18}$
# of POS (Noun, Verb, Adverb, Adjective)	$f_{19}, f_{20}, f_{21}, f_{22}$
# of words with nonzero score using Bing Liu's Opinion Lexicon	f_{23}
# of words with nonzero score using MPQA Subjectivity Lexicon	f_{24}
# of words with nonzero score using NRC Hashtag Sentiment Lexicon	f_{25}
# of words with nonzero score using Sentiment140 Lexicon	f_{26}
Maximum positive score for a token in the message using Bing Liu's Opinion Lexicon	f_{27}
Maximum positive score for a token in the message using MPQA Subjectivity Lexicon	f_{28}
Maximum positive score for a token in the message using NRC Hashtag Sentiment Lexicon	f_{29}
Maximum positive score for a token in the message using Sentiment140 Lexicon	f_{30}
Total score of the message using Bing Liu's Opinion Lexicon	f_{31}
Total score of the message using MPQA Subjectivity Lexicon	f_{32}
Total score of the message using NRC Hashtag Sentiment Lexicon	f_{33}
Total score of the message using Sentiment140 Lexicon	f_{34}

tive). We normalize the scores by diving each score by the scale (which is equal to 5) to obtain a score between -1 and +1. If a word is not directly found in the dictionary we retrieve all its synonyms from SENTIWORDNET. We then look for each of the synonyms in AFINN. If any synonym is found in AFINN, we assign the original word the same pleasantness score as its synonym. If none of the synonyms is present in AFINN, we perform a second level look up in the SENTIWORDNET dictionary to find synonyms of synonyms. If the word is present in SENTIWORDNET, we assign the score retrieved from SENTIWORDNET (between -1 and +1).

Table 3: Accuracy on 3-way classification task extending the baseline with additional features. All f_i refer to Table 2.

Model	F Measure			
	Positive Class	Negative Class	Neutral Class	Macro-Average
Baseline Model	36.93	30.66	15.38	33.79
+f_0	37.14	36.48	59.02	36.81
+ f_0 - f_1	63.73	47.19	66.24	55.46
+ f_0 - f_5	63.66	47.50	66.08	55.58
+ f_0 - f_7	63.58	47.55	66.08	55.56
+ f_0 - f_{11}	63.18	46.98	66.06	55.08
+ f_0 - f_{12}	63.14	48.52	65.75	55.83
+ f_0 - f_{15}	63.84	48.40	66.11	56.12
+ f_0 + f_{18}	64.42	48.57	66.30	56.50
+ f_0 - f_{22}	64.00	48.09	66.48	56.04
+ f_0 - f_{22} + Lexicon Based Features (f_{23} - f_{34})	67.50	52.26	66.57	59.83

4.4 Classification

After pre-processing and feature extraction we feed the features into a classifier. We tried various classifiers using the Scikit library [5]. After extensive experimentation it was found that SVM gave the best performance. The parameters of the model were computed using grid search. It was found that the model performed best with radial basis function kernel and 0.75 as the penalty parameter C of the error term. All the experimental are performed using these parameters for the model.

5 Results

In this section we present the experimental results for the classification task. We first present the score and rank obtained by the system on various test dataset followed by a discussion on the feature analysis for our system.

5.1 Overall Performance

The evaluation metric used in the competition is the macro-averaged F measure calculated over the positive and negative classes. Table 4 presents the overall performance of our system for different datasets.

5.2 Feature Analysis

Table 3 represents the results of the ablation experiment on the Twitter Test Data 2015. Using this ablation experiment, one can understand which features play an important role in identifying the sentiment of the tweet. It can be observed that the brown clusters plays an important role in determining the class of the tweet and improves the F-measure by around 20. Also, lexicon based features play a significant role by improving the F-measure by 3.

6 Conclusion

We presented results for sentiment analysis on Twitter by building a supervised system which combines lexicon based features with tweet specific features. We reported the overall accuracy for 3-way classification tasks: positive, negative and neutral. For our feature based approach, we perform feature analysis which reveals that the most important features are

Table 4: Overall Performance of the System.

Dataset	**Our Score**	**Best Score**	**Rank**
Twitter 2015	59.83	64.84	18
Twitter Sarcasm 2015	52.67	65.77	23
Twitter 2014	67.04	74.42	16
Twitter 2013	65.68	72.80	20
Twitter Sarcasm 2014	57.50	59.11	2
Live Journal 2014	69.91	75.34	21
SMS 2013	62.25	68.49	19

[5] http://scikit-learn.org/stable/modules/svm.html

those that combine the prior polarity of words and the lexicon based features. In the future, we will explore even richer linguistic analysis, for example, parsing, semantic analysis and topic modeling to improve our feature extraction component.

Acknowledgement

We thank Mayank Gupta and Arpit Jaiswal, International Institute of Information Technology, Hyderabad, India for assisting with the experiments as well as for interesting discussions on the subject. We would like to thank the SemEval 2015 shared task organizers for their support throughout this work. We would also like to thank the anonymous reviewers for their valuable comments.

References

Apoorv Agarwal, Fadi Biadsy, and Kathleen R. Mckeown. 2009. Contextual Phrase-level Polarity Analysis Using Lexical Affect Scoring and Syntactic N-grams. In *Proceedings of the* 12^{th} *Conference of the European Chapter of the ACL (EACL)*, pages 24–32.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment Analysis of Twitter Data. In *Proceedings of the Workshop on Languages in Social Media (LSM)*, pages 30–38.

Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, and Vasudeva Varma. 2012. Mining Sentiments from Tweets. In *Proceedings of the* 3^{rd} *Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 11–18.

Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the* 5^{th} *Conference on Language Resources and Evaluation (LREC*, pages 417–422.

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 *Proc. of the* 49^{th} *Annual Meeting of the ACL: Human Language Technologies (HLT)*, pages 42–47.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, pages 1–12.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the* 10^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177.

Bing Liu and Lei Zhang. 2012. A Survey of Opinion Mining and Sentiment Analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463.

Bing Liu. 2010. Sentiment Analysis and Subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*.

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* 7^{th} *International Workshop on Semantic Evaluation Exercises*, Atlanta, Georgia, USA.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *The* 2^{nd} *Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the* 7^{th} *International Workshop on Semantic Evaluation*, pages 312–320.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and A. Noah Smith. 2013. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 380–390.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the* 42^{nd} *Meeting of the ACL*, pages 271–278.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

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*, pages 73–80.

Sara Rosenthal, 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* 9^{th} *International Workshop on Semantic Evaluation*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT)*, pages 347–354.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-level Sentiment Analysis. *Computational Linguistics*, 35(3):399–433.