Sentiment Analysis in Twitter

Mayank Gupta, Ayushi Dalmia, Arpit Jaiswal and Chinthala Tharun Reddy
201101004, 201307565, 201305509, 201001069
IIIT Hyderabad, Hyderabad, AP, India
{mayank.gupta, arpitkumar.jaiswal, chinthalatharun.reddy}@students.iiit.ac.in, ayushi.dalmia@research.iiit.ac.in

***Abstract*—Microblogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life everyday in popular websites such as Twitter, Tumblr and Facebook. Spurred by this growth, companies and media organisation 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 ask peoples opinion on current debates. All this information can be obtained from microblogging services, as their users post their opinions on many aspects of their life regularly. In this work, we present a method which performs 3-class classification of tweet sentiment in Twitter [7]. We present an end to end system which can determine the sentiment of a tweet at two levels- phrase level and message level. We leverage the features of tweets to build the classifier which achieves an accuracy of 77.90% at phrase level and 58.13% at message level.**

***Keywords*—sentiment analysis, classification, twitter, SVM, feature**

I. Introduction

With enormous increase in web technologies, number of people expressing their views and opinions via web are increasing. This information is very useful for businesses, governments and individuals. With over 500+ million Tweets (short text messages) per day, Twitter is becoming a major source of information. Twitter is a micro-blogging site, which is popular because of its short text messages popularly known as Tweets. Tweets have a limit of 140 characters. Twitter has a user base of 240+ million active users and thus is a useful source of information. Users often discuss on current affairs and share their personals views on various subjects via tweets. Out of all the popular social medias like Facebook, Google+, Myspace and Twitter, we choose Twitter because of the following reasons:

- Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- Twitters 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 interests groups.
- Tweets are small in length, thus less ambigious
- Tweets are unbiased in nature

Using this social media we built models for classifying "tweets" into positive negative and neutral classes. We build models for two classification tasks: a 3-way classification of already demarcated phrases in a tweet into positive negative and neutral classes and another 3-way classification of entire message into positive negative and neutral classes. We experiment with the unigram model and feature based model. We do an incremental analysis of the features. We also experiment with a combination of models: combining unigram and feature based model.

For the phrase based classification task the unigram model achieves an accuracy of 62.24% which is 29% more than the chance baseline. The feature based model uses features and achieves an accuracy of 77.86%. The combination achieves an accuracy of 77.90% which outperform the baseline by 16%. For the phrase based classification task the unigram model achieves an accuracy of 51% which is 18% more than the chance baseline. The feature based model uses features and achieves an accuracy of 57.43%. The combination achieves and accuracy of 58.00% which outperform the baseline by 7%.

We use manually annotated Twitter data for our experiments. In this work we use three external resources: 1) a hand annotated dictionary for emoticons that maps emoticons to their polarity 2) an acronym dictionary collected from the web with English translations of over 5000 frequently used acronyms and 3)A lexicon which provides a prior score between -5 and +5 for commonly used English words and phrases.

The rest of the work is organised as follows. In section 2 we describe the dataset used for the task. Section 3 discusses the Resources and Tools followed by the explanation of Approach in Section 4 and Preprocessing steps in Section 4. Section 6 gives a detailed analysis of Experiments and Results and concludes in section 7.

II. Dataset

Twitter is a social networking and microblogging service that allows users to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging service, people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings. Following is a brief terminology associated with tweets.

- Emoticons: These are facial expressions: pictorially represented using punctuation and letters; they express the users mood
- Target: Users of Twitter use the @ symbol to refer to other users on the microblog. Referring to other users in this manner automatically alerts them.
- Hashtags: Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets.

In this project we use the dataset provided in SemEval 2013, Task 9 [3]. The dataset consists of tweet id's which are annotated with positive negative and neutral classes. The dataset is already divided into three sets: Training, Development and Testing. For sentiment analysis at the phrase level, the dataset contains 10586 phrases (which includes both the training and the development set) from different tweets and 4436 phrases from different tweets for testing purpose. Since, some of the tweets were not available while downloading, we are left with 8866 phrases for training and 3014 phrases for testing. For the second sub task, which is analysing the sentiment of the entire tweet we have 9684 tweets Ids (which includes both the training and the development set) and 3813 tweets Ids for testing. As mentioned before some of the tweets were not available while downloading. This leaves us with 9635 tweets for testing and 3005 tweets for testing.

III. Resources and Tools

In this work we use three external resources in order to preprocess the data and provide prior score for some of the commonly used words.

- Emoticon Dictionary: We use the emoticons list as given in [4] and manually annotate them. Table 1 is the snapshot of the dictionary. We categorise the emoticons into four classes: a) Extremely- Positive b) Positive c) Extremely- Negative d) Negative
- Acronym Dictionary: We crawl the website [1] 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 (discussed later). The acronym dictionary has 5297 entries. For example, asap has translation As soon as possible. Table 2 is the snapshot of the acronym dictionary.
- AFINN 111: AFINN [2] is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn rup Nielsen in 2009-2011. Table 3 is a snapshot of the AFINN dictionary.

We use the following tools in order to successfully implement our approach:

- Tweet Downloader [6] which is provided by the SemEval Task Organiser 2013. It contains a python script which downloads the tweets given the tweet id.

Table I. Emoticon Dictionary

Emoticon	Polarity
$:-):):o):]:3$	Positive
$:-D:D8DxDXD$	Extremely Positive
$:-/:/=/=</3$	Negative
$D:D8D=DXv.vDx$	Extremely Negative
$>:)B)B-):):-)>$	Neutral

Table II. Acronym Dictionary

Acronym	Expansion
admin	administrator
afaik	as far as I know
omg	oh my god
rol	rolling over laughing
wip	work in progress

- Tweet NLP [8], a twitter specific tweet tokeniser and tagger. It provides a fast and robust Java-based tokeniser and part-of-speech tagger for Twitter.
- LibSVM [5] is an integrated software for support vector classification. It supports multiclass classification.
- BeautifulSoup Python,the library provides an interface for crawling the web page. We use this for crawling the acronym dictionary.
- SciKit Python Library, for Naive Bayes classifier
- Svmutil Python Library, for implementing SVM

IV. Preprocessing

A. Tokenisation

After downloading the tweets using the tweet id's provided in the dataset, we first tokenise the tweets. This is done using the Tweet-NLP developed by ARK Social Media Search. This tool tokenises the 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 in the sense it tags the twitter specific entries like Emoticons, Hashtag and Mentions too. After obtaining the tokenised and tagged tweet we move to the next step of preprocessing.

B. Remove Non-English Tweets

Twitter allows more the 60 languages. However, this work currently focuses on English tweets only. We remove the tweets which are non-English in nature. This is because, the tokeniser we use is compatible only with English tweet text.

Table III. AFINN-111 Dictionary

Word	Score
adore	3
aggressive	-2
bitch	-5
breathtaking	5
celebrate	3

C. Replacing Emoticons

Emoticons play and 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.

D. Remove Url

The urls which are present in the tweet are shortened using TinyUrl due to the short nature of Tweet messages. These shortened Url's did not carry much information regarding the sentiment of the tweet. Thus these are removed.

E. Remove Target

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

F. Replace Negative Mentions

Tweets consisting of various notions of negation. In general, words ending with 'ńt' are appended with a not. Before we remove the stopwords 'not' is replaced by the word 'negation'. Negation play a very important role in determining the sentiment of the tweet. This is discussed later in detail.

G. Hashtags

Hashtags are basically summariser 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 it as a feature.

H. Sequence of Repeated Characters

Twitter provides a platform for users to express their opinion in an informal way. Tweets are written in random form, without any focus given to 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 emphasise 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 'cool' and 'cooooool'.

I. Numbers

Numbers are of no use when measuring sentiment. Thus, numbers which are obtained as tokenised unit from the tokeniser are removed in order to refine the tweet content.

J. Nouns and Prepositions

Given a tweet token, we identify the word as a Noun word by looking at its part of speech tag given by the tokeniser. If the majority sense (most commonly used sense) of that word is Noun, we discard the word. Noun words dont carry sentiment and thus are of no use in our experiments. The same reasoning go for prepositions too.

K. Stop-word Removal

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. And also, stop words dont carry any sentiment information and thus are of no use to us. We create a list of stop words like he, she, at, on, a, the, etc. and ignore them while scoring the sentiment.

V. Approach

In message based sentiment analysis we build unigram model and feature based model. We also try to perform classification using a combination of both these models. Our approach can be divided into various steps. Each of these steps are independent of the other but important at the same time. Figure 1 and 2 represent the approach for training and testing the model.

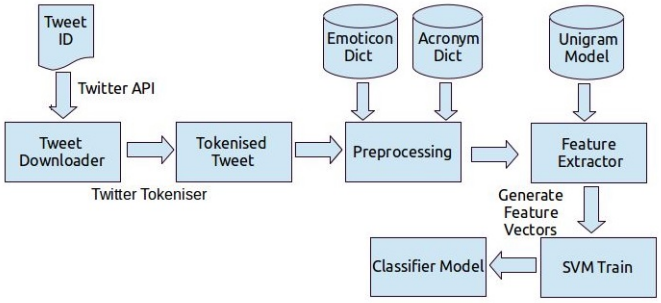

Figure 1. Flow Diagram of Training: Hybrid Model

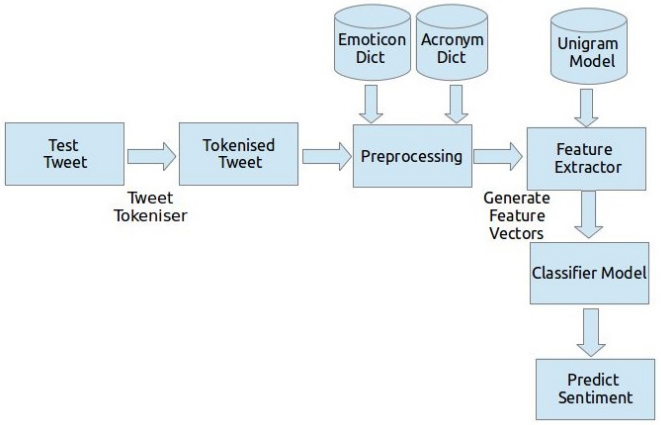

Figure 2. Flow Diagram of Testing: Hybrid Model

A. Baseline Model

In the baseline approach, we first clean the tweets. We perform the preprocessing steps listed in section 4 and learn the positive & negative frequencies of unigrams, bigrams and trigrams in training. Every token is given three probability scores: Positive Probability (Pp, Negative Probability (Np) and Neutral Probability (NEp).

Table IV. UNIGRAM BIGRAM AND TRIGRAMS

Unigram	Bigram	Trigram
bad	negation wait	can negation wait
really	laugh loud	anywhere anytime great
win	looking forward	negation wait !
shit	goodnite luck	wait eyes stye
laugh	negation miss	
fun	cant wait	
loud	love !	
thanks	goodnite morning	
fuck		
looking		
forward		
amazing		

$$P_f = \textit{Frequency in Positive Training Set}$$

$$N_f = \textit{Frequency in Negative Training Set}$$

$$NE_f = \textit{Frequency in Neutral Training Set}$$

$$P_p = \textit{Positive Probability} = P_f/(P_f + N_f + NE_f)$$

$$N_p = \textit{Negative Probability} = N_f/(P_f + N_f + NE_f)$$

$$NE_p = \textit{Neutral Probability} = NE_f/(P_f + N_f + NE_f)$$

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 or negative sentiment with high confidence. We call such words, **Emotion Determiner**. A token is considered Emotion Determiner using something similar to the theory of *Triangular Inequality*. The probability of emotion for any one sentiment must be greater than or equal to the probability of the other two sentiments. by a certain threshold (t_{ed}). It is found that we have different thresholds for unigrams, bigrams and trigrams. The parameter for the three tokens is tuned and the optimal threshold values are found. Note, before calculating the probability values, we filter out those tokens which are infrequent (appear in less than 10 tweets). Table 4 shows a list of unigrams, bigrams and trigrams which obey the minimum optimal threshold criteria.

B. Feature Based Model

As in the previous model, in this model too we perform the same set of preprocessing techniques as mentioned in section 4.

1) Prior Polarity Scoring: A number of our features are based on prior polarity of words. For obtaining the prior polarity of words, we use AFINN dictionary and extend it using senti-Wordnet. We first look up tokens in the tweet in the AFINN lexicon. This dictionary of about X English language words assigns every word a pleasantness score between -5 (Negative) and +5 (Positive). We first normalize the scores by diving each score by the scale (which is equal to 5). If a word is not directly found in the dictionary we retrieve all synonyms from Wordnet. 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 senti-Wordnet dictionary. If the word is present in senti-Wordnet, we assign the score retrieved from senti-Wordnet (between -1 and +1). In this way we could find the score of X% of the tokens.

Table V. N REFERS TO SET OF FEATURES WHOSE VALUE IS A POSITIVE INTEGER. THEY ARE PRIMARILY COUNT FEATURES; FOR EXAMPLE, COUNT OF NUMBER OF POS, COUNT OF CAPITALISED, COUNT OF POSITIVE AND NEGATIVE EMOTICONS. R REFERS TO FEATURES WHOSE VALUE IS A REAL NUMBER; FOR EXAMPLE, SUM OF THE PRIOR POLARITY SCORES OF WORDS WITH PART-OF-SPEECH OF ADJECTIVE/ADVERB/VERB/NOUN, AND SUM OF PRIOR POLARITY SCORES OF ALL WORDS. B REFERS TO THE SET OF FEATURES THAT HAVE A BOOLEAN VALUE; FOR EXAMPLE, PRESENCE OF CAPITALIZED TEXT.

Feature Description	Feature Id	Feature Type
Polarity Score of the Tweet	f_1	R
Percentage of Capitalised Words	f_2	R
# of Positive Capitalised Words	f_3	N
# of Negative Capitalised Words	f_4	N
Presence of Capitalised Words	f_5	B
# of Positive Hashtags	f_6	N
# of Negative Hashtags	f_7	N
# of Positive Emoticons	f_8	N
# of Extremely Positive Emoticons	f_9	N
# of Negative Emoticons	f_{10}	N
# of Extremely Negative Emoticons	f_{11}	N
# of Negation	f_{12}	N
Positive POS Tags Score	f_{13}	R
Negative POS Tags Score	f_{14}	R
Total POS Tags Score	f_{15}	R
# of special characters like ? ! and *	$f_{16}.f_{17}, f_{18}$	N
# of POS	$f_{19}, f_{20}, f_{21}, f_{22}$	N

2) Features: We propose a set of features listed in Table X for our experiments. These are a total of X type of features. We calculate these features for the whole tweet in case of message based sentiment analysis and for the extended phrase (obtained by taking 2 tokens on either sides of the demarcated phrase) in case of phrase based sentiment analysis. We refer to these features as Emotion-features throughout the paper. Our features can be divided into three broad categories: ones that are primarily counts of various features and therefore the value of the feature is a natural number *N*. Second, features whose value is a real number *R*. These are primarily features that capture the score retrieved from AFINN. Thirdly, features whose values are boolean *B*. These are bag of words, presence of exclamation marks and capitalized text. Table 5 summarises the features used in our experiment.

VI. EXPERIMENTS AND RESULTS

- Positive versus Negative versus Neutral for Phrase Level Sentiment Analysis
- Positive versus Negative versus Neutral for Message Level Sentiment Analysis

For each of the classification tasks we present two models, as well as results for the combinations of these models:

- Baseline Model
- Feature Based Model
- Baseline plus Feature Based Model

For the Baseline plus Feature Based Model, we present feature analysis to gain insight about what kinds of features are adding most value to the model.

Table VI. PARAMETER TUNING FOR BASELINE

Unigram	Bigram	Accuracy
0.4	0.4	47.48%
0.5	0.5	47.98%
0.6	0.6	48.41
0.7	0.7	50.78%
0.8	0.8	48.61%
0.7	0.8	51.31%
0.7	0.9	50.74%
0.7	0.8	**51.81%**
0.7	1	51.08%
0.7	1	51.68%

Experimental-Set-up: For all our experiments we use Support Vector Machines (SVM)

A. Phrase Level Sentiment Analysis

For phrase level sentiment analysis the major challenge was to identify the sentiment of the tweet pertaining to the context of the tweet. We know that tokens can represent different aspects in different contexts. In order to capture this sentiment, we extend the phrase on either side by size two. That is given a phrase and the tweet in which it belongs, we extract the phrase which includes tokens on either side of the phrase. We believe that this helps in taking into consideration the context of the tweet. But after experimentation it was found that the accuracy of the system dropped for both the models. Therefore we only use the phrases as demarcated to predict the sentiment of the tweet.

1) Baseline Model: For the baseline model, we only consider the unigrams and bigrams. Taking the trigrams leads to drop in accuracy. We perform the parameter tuning for threshold (t_{ed}). It is listed in Table 6. It is found that the system performs best for thresholds X and Y for unigram and bigram respectively. The accuracy achieved for the baseline model is 62.24%. This is a hard line achieving 29% more than the chance probability.

2) Feature Based Model: For the feature based model we used the features as listed in Table 5. The model is trained using the features. We create feature vectors for the test samples and feed it to the model. The accuracy achieved using all the features is 77.86%.

3) Baseline plus Feature Based Model: Table 7 presents classifier accuracy when features are added incrementally. We start with our baseline model and subsequently add various sets of features. First, we add polarity score (rows f_1) in Table 5) and observe a gain in 15% performance. Capitalisation Features also do not help much (rows f_2,f_3,f_4). Next, we add all hashtag based features (rows f_6, f_7) and observe no improvement. Similary with addition of emoticons (rows f_8,f_9,f_{10}, f_{11}) and find no improvement. This is probably due to the short nature of the phrases, emoticons and hashtags are not major contributors for improving the accuracy of the classifier. We see an additional increase in accuracy by 0.03% when we add negation (rows f_{12}). The accuracy jumps to 77.90% when we add prior polarity score of POS tags (rows f_{16}, f_{17}, f_{18}). Adding special characters (rows f_{16},f_{17},f_{18}) drops the accuracy to 77.50%. Next, adding capitalisation features (rows f_{19},f_{20},f_{21}, f_{22}) improves the accuracy by 0.46%. From these experiments we conclude that the most important features are those that involve prior polarity of POS tags. All other features play a marginal role in achieving the best performing system.

Table VII. ACCURACY FOR 3-WAY CLASSIFICATION TASK USING BASELINE AND FEATURES. ALL f_i REFER TO TABLE 5

Baseline Model	62.24%
Baseline Model + f_1	77.10%
Baseline Model + f_1+ f_2 f_3 + f_4	77.10%
Baseline Model + f_1 f_6 + f_7	77.10%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 + f_{10} + f_{11}	77.10%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12}	77.13%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15}	**77.90%**
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15} + f_{16} + f_{17} f_{18}	77.50%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15} + f_{16} + f_{17} f_{18} + f_{19} + f_{20} + f_{21} f_{22}	77.86%

B. Message Level Sentiment Analysis

For message level sentiment analysis the most difficult part was to resolve ambiguity. A message can contain both positive and negative sentiments and hence it is difficult to determine the stronger sentiment in the tweet. As a result the highest accuracy achieved is also not at par with the phrase based sentiment analysis. For message based sentiment analysis, the best accuracy achieved is 58.13%

1) Baseline Model: For the baseline model, we consider the unigrams bigrams and trigrams. We perform the parameter tuning for threshold (t_{ed}). It is listed in Table 8. It is found that the system performs best for thresholds 0.7, 0.9 and 0.8 for unigram, bigram and trigram respectively. The accuracy achieved for the baseline model is 51.81%. This is a hard line achieving 18% more than the chance probability.

2) Feature Based Model: For the feature based model we used the features as listed in Table 5. The model is trained using the features. We create feature vectors for the test samples and feed it to the model. The accuracy achieved using all the features is 57.43%.

3) Baseline plus Feature Based Model: Table 7 presents classifier accuracy when features are added incrementally. We start with our baseline model and subsequently add various

Table VIII. PARAMETER TUNING FOR BASELINE

Unigram	Bigram	Trigram	Accuracy
0.4	0.4	0.4	58.68%
0.5	0.5	0.5	58.94%
0.6	0.6	0.6	59.41
0.7	0.7	0.7	59.78%
0.8	0.8	0.8	58.61%
0.7	0.8	0.8	60.90%
0.7	0.9	0.9	61.74%
0.7	0.8	0.9	**62.24%**
0.7	1	1	61.86%
0.7	1	0.9	62.01%

sets of features. First, we add polarity score (rows f_1) in Table 5) and observe a gain in 3% performance. Capitalisation Features also do not help much (rows f_2,f_3,f_4) and improves only by 0.12%. Next, we add all hashtag based features (rows f_6, f_7) and observe no improvement. With addition of emoticons (rows f_8,f_9,f_{10}, f_{11}) we find minor increment in the performance by 0.04%. We see an additional increase in accuracy by 2% when we add negation (rows f_{12}). Thus negation plays an important role in improving the accuracy of the classifier in a substantial way. The accuracy jumps to 57.77% when we add prior polarity score of POS tags (rows f_{16}, f_{17}, f_{18}). Adding special characters (rows f_{16},f_{17},f_{18}) improves the accuracy to 58.10%. Next, adding capitalisation features (rows f_{19},f_{20},f_{21}, f_{22}) drops the accuracy by 0.10%. From these experiments we conclude that the most important features are those that involve prior polarity of POS tags. All other features play a marginal role in achieving the best performing system.

VII. CONCLUSION

We presented results for sentiment analysis on Twitter. We use a baseline model and report an overall the accuracy 3-way classification tasks: positive versus negative versus neutral. We presented a comprehensive set of experiments for two level classification: message level and phrase level on manually annotated data that is a random sample of stream of tweets. We investigated two kinds of models: Baseline and feature based models and demonstrate that combination of both these models perform the best. For our feature-based approach, we do feature analysis which reveals that the most important features are those that combine the prior polarity of words and their parts-of-speech tags. In future work, we will explore even richer linguistic analysis, for example, parsing, semantic analysis and topic modeling.

Table IX. ACCURACY FOR 3-WAY CLASSIFICATION TASK USING BASELINE AND FEATURES. ALL f_i REFER TO TABLE 5

Baseline Model	51.81%
Baseline Model + f_1	54.17%
Baseline Model + f_1+ f_2 f_3 + f_4	54.30%
Baseline Model + f_1 f_6 + f_7	54.30%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 + f_{10} + f_{11}	54.34%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12}	56.50%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15}	57.77%
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15} + f_{16} + f_{17} f_{18}	**58.10%**
Baseline Model + f_1 f_6 + f_7 + f_8 + f_9 f_{10} + f_{11} + f_{12} + f_{13} f_{14} + f_{15} + f_{16} + f_{17} f_{18} + f_{19} + f_{20} + f_{21} f_{22}	58.00%

REFERENCES

[1] Acronym list. [Online]. Available: http://www.noslang.com/dictionary/

[2] Afinn-111. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[3] Dataset. [Online]. Available: http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools

[4] Emoticon list. [Online]. Available: http://en.wikipedia.org/wiki/List_of_emoticons

[5] Libsvm. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[6] Tweet downloader. [Online]. Available: http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools

[7] Twitter. [Online]. Available: http://twitter.com

[8] Twitter nlp. [Online]. Available: https://github.com/brendano/ark-tweet-nlp/tree/master/src/cmu/arktweetnlp