2013 IEEE 25th International Conference on Tools with Artificial Intelligence

Aspect-based Twitter Sentiment Classification

Hsiang Hui Lek and Danny C.C. Poo
Department of Information Systems
School of Computing, National University of Singapore
E-mail: {lekhsian, dpoo}@comp.nus.edu.sg

Abstract— **Due to the popularity of Twitter, sentiment classification for Twitter has become a hot research topic. Previous studies have approached the problem as a tweet-level classification task where each tweet is classified as positive, negative or neutral. However, getting an overall sentiment might not be useful to organizations which are using twitter for monitoring consumer opinion of their products/services. Instead, it is more useful to determine specifically which aspects of the products/services the users are happy or unhappy about. This paper proposes an aspect-based sentiment classification approach to analyze sentiments for tweets. To the best of our knowledge, we are the first to perform sentiment analysis for Twitter in this manner. We conducted several experiments and show that by incorporating results from the aspect-based sentiment classifier, we are able to improve existing tweet-level classifiers. The experimental results also demonstrated that our approach outperforms existing state-of-the-art approaches.**

Sentiment Analysis; Opinion Mining; Twitter Sentiment Analysis; Aspect-based Sentiment Analysis

I. INTRODUCTION

Twitter has become one of the major social media websites today. The huge amount of tweets and availability of official APIs for retrieving tweets have made it an attractive resource for organizations to monitor users' opinion. For example, Twitter has been used to analyze users' sentiment during elections [22, 24] and used to predict stock/financial market [3, 14].

Previous studies have approached this sentiment analysis problem as a tweet-level sentiment classification task similar to that of document-level sentiment classification which is often used to classify movie reviews [16]. Tweet-level sentiment classification determines the overall sentiment orientation of a tweet. However, getting an overall positive or negative sentiment might not be useful to the organizations as it is more important to determine what exactly their consumers are happy or unhappy about. Furthermore, even though Twitter restricts each tweet to a maximum of 140 characters, tweets can still contain mixed sentiments about a service/organization. Consider the following tweet concerning a telecommunication company, "*No more turtle speed 3G from StarHub! LTE is so cool~*", the author suggests that the 3G connectivity is bad but the LTE connectivity offered by the company is good. Assigning a positive or negative polarity to this tweet will be wrong and assigning a neutral polarity will result in loss of information. It would be more ideal to extract the important aspects and their associated polarity. In this case, *[3G, turtle speed, -]* and *[LTE, cool, +]* (in the form of *[aspect, sentiment words, polarity]*).

This paper proposes an approach to perform aspect-based sentiment classification for Twitter. To the best of our knowledge, we are the first to perform sentiment classification for Twitter in this manner. We conducted several experiments on publicly available Twitter datasets and show that our aspect-based Twitter sentiment classifier is able to boost the performance of existing tweet-level sentiment classifiers. Thus achieve more accurate sentiment classification than the state-of-the-art approaches.

II. RELATED WORK

Early work in sentiment analysis mostly considers movie reviews [11, 16] and product reviews [9, 17, 18, 26].

Review data is relatively easier to work with since sentences in reviews are more grammatically correct compared to tweets. Tweets on the other hand are short, informal, colloquial, and can contain slangs and abbreviations. Thus, many of the existing natural language processing tools such as Part-of-Speech (POS) tagger, tokenizer, and dependency parser fail to work well because they are typically trained on newswire data. Instead, current Twitter sentiment analysis approaches [2, 5, 8, 19, 20] adopt a machine learning approach similar to text classification.

Training data are often obtained by using noisy labels (also known as distant supervision). This allows us to classify the tweets without manual supervision. Go et al. [8] and Read [19] exploit emoticons such as ":-)" and ":-(" to label the tweets positive and negative respectively. They then treat the problem as a text classification task and use machine learning techniques such as Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machines (SVM) to train a classifier. Barbosa and Feng [2] on the other hand, construct their training data from a few different Twitter sentiment analysis websites.

Features for training classifiers are usually content-based or lexical-based. Go et al. [8] experiment with content-based features such as unigram, bigram, and POS features on NB,

1082-3409/13 $31.00 © 2013 IEEE
DOI 10.1109/ICTAI.2013.62

MaxEnt, and SVM classifiers. They managed to obtain a classification accuracy of 83% by using MaxEnt classifier trained on unigram and bigram. Twitter hashtags and emoticons have also been used as content-based features [5].

Lexical-based features on the other hand, include the use of external resources or lexicons. Sentiment lexicons have often been used to determine the subjectivity and polarity of words. Some of the commonly used lexicons include the subjectivity lexicon [25], SentiWordNet [1], and opinion lexicon [13]. The polarity information is then incorporated into the classifier. For example, Speriosu et al. [21] adopt a label propagation approach which utilizes the subjectivity lexicon [25]. Nodes are created for every word in the lexicon, and each tweet is connected to these nodes by edges if the tweet contains these words. Apart from using sentiment lexicon, Saif et al. [20] incorporate semantic features for sentiment classification. They take advantage of third-party services to extract entities from tweets. The entities are then mapped to semantic concepts and incorporated as features in the classifier.

The approaches discussed so far are target-independent and do not take into account the target of interest. However, as illustrated in the previous section, a tweet can contain mixed sentiments and multiple targets. Thus a target-independent approach might not classify a tweet accurately. Jiang et al. [10] adopt a target-dependent approach. On top of the target-independent features mentioned above, they incorporated syntactic features from a dependency parser. Given a tweet and a target, they generate features out of tokens which are syntactically associated with the target. These features include the transitive verb, intransitive verb, adjective, and adverb associated with the target. Unlike other approaches which we have discussed so far, this is a supervised approach which cannot use noisy labels to obtain training data. Instead, the training data has to be manually labeled with respect to a specific target.

Previous work [9, 12, 17, 18, 26] that consider aspect-based sentiment analysis typically work with product reviews. Hu and Liu [9] extract noun phrases and use associative mining to identify the frequent noun phrases as product aspect candidates. The adjective adjacent to an aspect candidate is then treated as the corresponding sentiment word. Popescu and Etzioni [17] extract noun phrases and evaluate each noun phrase by computing the PMI [23] between the noun phrases and various meronymy discriminators (part-of expressions). A list of dependency rule templates is then used to identify the corresponding sentiment words. The semantic orientations of sentiment words are then determined using SO-PMI [23].

III. Aspect-based Sentiment Classifier For Twitter

Similar to aspect-based sentiment analysis for product reviews, given a tweet, we aim to extract a list of aspects and their corresponding sentiments. Consider the following tweet "*Rain is not an excuse for the super horrible reception today Singtel..*", our system produces a list of aspects along with their associated sentiment words and polarity: *[reception, horrible, -]*. Formally, this process consists of three main steps:

1. **Aspect-sentiment extraction.** Given a tweet, this step determines a list of possible aspect candidates along with their associated sentiments and polarity
2. **Aspect ranking and selection.** A tweet can express many different opinions. Only those important aspects should be selected. For example, when classifying tweets on a telecommunication company, some of the aspects of interest include *customer service*, *3G connectivity*, *speed,* etc. In this step, the aspect candidates are then ranked and the set of most significant aspects are selected as the expected aspects.
3. **Aspect classification.** Using the set of expected aspects and results from the aspect-sentiment extraction step, we obtain the final list of aspects along with their polarity for each tweet.

A. Aspect-Sentiment Extraction

This step takes in a tweet and extracts a list of aspects along with their associated sentiments and polarity. Similar to previous work in aspect-based sentiment extraction for product reviews, we consider each noun in a tweet to be a possible aspect candidate. In order to determine the associated sentiment, most of the previous work in aspect-based sentiment analysis for product reviews makes use of results from a dependency parser [12, 17, 18] or a sentiment lexicon [9]. We show in section VI that a linguistic approach which makes use of an existing dependency parser does not work well with tweets. Nevertheless, the experiments show that linguistic information such as POS tags is still useful.

Our extractor makes use of a POS tagger, a sentiment lexicon, and a few gazetteer lists to obtain the results in the form of *[aspect, sentiment words, polarity]*. The sentiment words in the tuple are used to classify the polarity of the aspect. We employ the Twitter POS tagger from CMU [7] which is especially designed for Twitter data. Unlike most POS taggers which are based on the Penn Treebank tags, a set of specialized tags is used to annotate the tweets such as E (emotion), @ (at-mention), U (URL), N (noun), V (verb), and A (adjective). For the sentiment lexicon, we use the opinion lexicon from [13]. Our experiment shows that this lexicon produces better results compared to the subjectivity lexicon [25] and SentiWordNet [1] for Twitter data. The sentiment lexicon is used to determine the polarity of the sentiment words. For the gazetteer lists, we utilize a stop word list, a swear word list, and an intensifier word list. Aspect candidates which are found in the stop word list are ignored. A word is classified as negative if it is found in the swear word list. The intensifier word list contains words like "*very*", "*extremely*", "*bloody*", "*damn*", etc. This is important because some of the words such as "*damn*", "*bloody*", and "*fucking*" might be erroneously be considered as sentiment words but they are only used to intensify the opinion words following them.

The following describes the steps for extracting the results from a tweet:

1. Tokenization, POS-tagging, and sentence extraction are first performed on the tweet. Subsequent steps are done on a sentence level.
2. For sentences which contain "*but*", we only consider the tokens after "*but*". This is because the pair of clauses before and after the coordinating conjunction ("*but*") usually takes on different orientation, however very often, the author tends to emphasize more on the latter clause. For example, consider the tweet *"There is free channels but no nice shows to watch"* even though we might extract two tuples: *[channel, free, +]* and *[show, nice-neg, -]* (*-neg* denotes negation), the latter should take precedence over the former.
3. Each noun, abbreviation, @mention, or hashtag is treated as a possible aspect candidate.
4. For each aspect candidate, we find the closest verb to its left. If this verb can be found in the sentiment lexicon/swear word list, a tuple of the form *[aspect, verb, polarity(verb)]* is added to the result list.
5. We scan the left of the aspect candidate again to find the closest adjective, adverb, or hashtag that is contained in the sentiment lexicon/swear word list. Similarly, *[aspect, token, polarity(token)]* is added to the result list when there is a match. To handle the comparison cases, if the token is a comparative/superlative adjective or adverb, the polarity of the token is switched. For example, consider the tweet *"iPhone hotspot is better than home internet."*, since "*better*" is a positive sentiment word, we assign a negative polarity to *home internet* since it usually suggests that the latter clause is not as good/bad compared to the former.
6. Similar to step 4 and 5, we scan the right of the aspect candidate for the closest verb and adjective/adverb/hashtag. The tuples are then added to the result list accordingly. Unlike step 4 and 5, two different cases can happen. Firstly, a verb can be copulative and should be ignored. Secondly, a word might be an intensifier to the next following sentiment word and should also be ignored. For example, *"The movie is damn good"*, if we consider *"damn"* as the associating sentiment word, we would have incorrectly classified the aspect as negative instead of positive.
7. Note that we also consider negation cases and toggle the polarity of a word if a negation word is encountered within a negation window size. This is set to be 6 in our experiments. Furthermore, if a word cannot be found in the sentiment lexicon/swear word list, we correct the word progressively in an attempt to find a match. The correction steps include stemming, reducing the repeated letters if the word contains more than two repeated letters (for example *"looovvveee"* is corrected as *"love"*), and spelling correction.

B. Aspect Ranking and Selection

Oftentimes tweets are retrieved from the Twitter search API using a query. These tweets tend to refer to a specific domain of interest so it is more important to determine the set of important aspects. To obtain this set of aspects, an aspect ranking and selection step is performed.

This step first takes in a collection of tweets in a particular domain of interest and the query (or target) used to retrieve the tweets. Next, the aspect-sentiment extraction algorithm described in the previous section is performed on every tweet and the results are combined. We count the number of times each aspect appears in this list and rank the aspect in decreasing order. Aspects which fall below a threshold count are ignored. From this filtered set of aspects, we employ a Pointwise Mutual Information (PMI) approach similar to [23] to select the list of most significant aspects that are similar to the target. The PMI value of a pair of aspect-target is calculated (Equation 1) and the aspect is selected if it exceeds a certain PMI value.

$$PMI(p,q) = \log_2\left(\frac{\text{hits}(q \text{ AND } q)}{\text{hits}(p).\text{hits}(q)}\right) \quad (1)$$

where *hits(query)* = number of hits returned by the Google search engine for a given query.

C. Aspect Classification

The aspect classification step considers aspects which are found in the set of important aspects (obtained from the previous step). We then count the number of times the aspect is classified as positive or negative. If there is equal number of positive and negative counts, the aspect is ignored. Otherwise, the aspect is assigned as positive or negative based on the predominant count. Since tweets are generally short, most of the counts tends to be small (<= 3).

IV. Mining Sentiment Expressions to Further Improve the Aspect-based Sentiment Classifier

As our approach is very dependent on the sentiment lexicon and the word lists, we also propose an enhancement step which will mine target-dependent sentiment expressions. The sentiment expressions extracted are similar to [4] except that each expression starts with or ends with the TARGET token which denotes the target. Table I shows some examples of these sentiment expressions. The sentiment expressions allow us to handle words which not found in the sentiment lexicon as well as multi-word sentiments.

We define sentiment expressions as a combination of a target together with the verb phrase or adjective before or

TABLE I. Examples of Sentiment Expressions

Sentiment Expressions	Polarity
thank-TARGET	Positive
TARGET-ftw	Positive
TARGET-COP-amazing	Positive
hang-out-with-TARGET	Positive
sad-TARGET	Negative
will-miss-TARGET	Negative
TARGET-screw-up	Negative
so-sorry-to-hear-about-TARGET	Negative

after the target. The next step is then to make use of a large corpus that is annotated with the target along with its polarity to mine and classify these expressions. However, such a corpus is difficult to obtain. Instead, we use the training dataset from the Stanford Twitter Sentiment (STS) [8] which consists of 1.6 million tweets. The training dataset was obtained and classified using noisy labels. We treat every noun in the tweet as a possible target. For each tweet, we extract the sentiment expressions associated with each target (copular verbs are replaced with COP). We then count the number of times an expression appears as positive and negative tweets. The expressions are extracted without negations. Expressions which are preceded with a negation word are considered as found in a tweet of the opposite polarity.

We tried two methods of assigning the polarity to the expressions. The first method is a probabilistic method which estimates the positivity/negativity of an expression based on its frequencies of occurrence. The second method is a graph-based method which takes into account the co-occurrence relationship of the expressions when assigning the polarity to these expressions.

A. Probabilistic Polarity Assignment

This is a simple method which counts the number of times an expression appears in the positive or negative tweets from the training data. The polarity is then assigned based on equations 2, 3, and 4. In our experiment, we set M1 to be 3.

$$C_t(e) = C_p(e) + C_n(e) \quad (2)$$

$$P(positive \mid e) = C_p(e) / C_t(e) \quad (3)$$

$$polarity(e) = \begin{cases} positive \text{ if } C_t(e) >= M1 \text{ and } P(positive \mid e) >= 0.7 \\ negative \text{ if } C_t(e) >= M1 \text{ and } P(positive \mid e) <= 0.3 \\ x \text{ otherwise} \end{cases} \quad (4)$$

where $C_p(e)$ = number of times expression e appears in positive tweets, $C_n(e)$ = number of times expression e appears in negative tweets.

B. Graph-based Polarity Assignment

It is noted that for each tweet, there can be more than one expression extracted. Many of these expressions co-occur together and are likely to take on the same polarity. Since we extract expressions without negations, we can also determine expressions which are likely to be of opposite polarity. Based on this idea, we can then construct a graph with the sentiment expressions as nodes (Figure 1). Each node in the graph denotes a sentiment expression. Nodes are connected to each other by two kinds of lines. Solid line indicates that two expressions co-occur together (direct association) in the tweets while the dash line indicates that two expressions co-occur in the tweets but one of them is preceded with a negation word (opposite association). A weight is also assigned to each edge which determines the strength of the association. This weight is calculated based on the number of times two sentiment expressions co-occur together. They are given a positive value for direct association and negative value for opposite association. The weights are summed up for cases where there are both direct and opposite

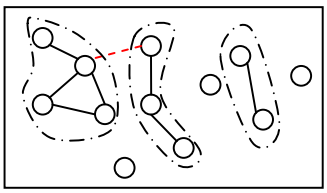

Figure 1. An example graph of the sentiment expressions

associations (and removed if the final value is 0). The weight are normalized by taking the logarithm of its absolute value (1 is added to weights whose absolute value are 1 to ensure that the log value is non-zero).

We can then group up the nodes together by clustering. Nodes in the same cluster are given the same polarity. To do this, we make use of the affinity propagation clustering technique [6], a recent state-of-the-art clustering method. The weight of each edge is then used as the similarity values between a pair of nodes. Out of the 48848 expressions and 303545 edges, 28538 clusters are formed. The largest cluster consists of 386 nodes. Figure 2 shows a small snapshot of the cluster size. We see that the size of the clusters follows a long tail behavior where there are very few large clusters and many small clusters. 86.7% of the clusters consist of only an isolated node. Equation 5 is then used to determine the positivity of a cluster and equation 4 is used to assign the polarity of the cluster (instead of an expression).

$$P(positive \mid cluster) = \frac{1}{n}\sum_{i=0}^{n} P(positive \mid e_i) \quad (5)$$

where n is the size of the cluster

We did a comparison between the two methods and notice that even though the graph-based method considers the co-occurrence relationship of the expressions, it does not result in a better polarity assignment. Conversely, a simple probabilistic approach allows us to have a more accurate polarity assignment. Thus it is used to assign the polarity of the sentiment expressions in our experiments.

Out of the 48848 sentiment expressions, 12805 (11642) of the expressions are classified as positive (negative). Table II shows the distribution of N-grams of the sentiment expressions (ignoring the TARGET tag) for all the expressions and those that are positive or negative.

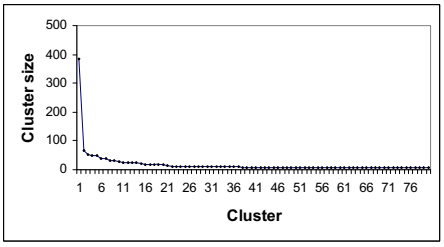

Figure 2. Small snapshot of the cluster size

TABLE II. DISTRIBUTION OF N-GRAMS OF THE EXPRESSIONS

N-Gram	1	2	3	4	>=5
All	8624	22291	13561	3635	737
Pos/Neg	3747	10592	7452	2157	499

V. TWEET-LEVEL SENTIMENT CLASSIFICATION

We conducted several experiments and show that our aspect-based classifier can improve the performance of the conventional tweet-level classifier. This section describes our tweet-level sentiment classifier and the features used. We experimented with MaxEnt and NB classifiers using the MALLET [15] package. The experimental results also show that our tweet-level sentiment classifier is comparable with the state-of-the-art implementations.

The tweets are first preprocessed where some tokens such as URL, numbers and emails are replaced with special tokens. The classifier uses both content and lexical features. The content features include unigram, bigram, and emoticon. For lexical features, we use the opinion lexicon [13] and the swear word list to detect whether a token is positive or negative. Similar to the aspect-based sentiment classifier, we toggle the polarity of a word if a negation word is detected within a negation window size of 6.

A. Incorporating Aspect-based Sentiment Classifier to existing Classifier

We tried three methods to incorporate results from the aspect-based sentiment classifier into the tweet-level sentiment classifier:

Layered classification. For a given tweet along with its target, we first perform classification using the aspect-based sentiment classifier (described in section III) to obtain a list of aspects and their polarity. A tweet is assigned as positive (negative) if the percentage of positive (negative) aspects exceeds a threshold value of 0.85. For tweets where we are unable to obtain a polarity classification, we then use the tweet-level classifier to determine the final polarity. The motivation of this approach is that since the aspect-based sentiment classifier is able to capture the explicit sentiment associated with a target, it is expected to produce highly precise classification.

Incorporate the sentiment words as features. Since our aspect-based sentiment classifier also returns the sentiment words associated with an aspect, we could incorporate this as a feature (in the form of TARGET#sentiment_words) into the tweet-level classifier. This idea is similar to [10] except that we do not differentiate between how the opinion is associated with the aspect. For example, suppose a tweet "*nice phone*" generates the tuple *[phone, nice, +]*, and another tweet "*the phone is nice*" also generates the same tuple, they will be treated the same and the feature TARGET#nice is generated.

Incorporate the polarities as features. To further tackle the problem of sparsity of target-dependent features, we also tried a less specific variant that back off to only the polarity. Features generated are in the form TARGET#polarity (e.g. TARGET#+).

For the second and third method, to overcome sparsity of the target-dependent features, we follow the approach described in [10]: a special binary feature is included which is set to 1 if the target-dependent features (described above) are present or 0 otherwise.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Datasets

We evaluate our approach with four datasets. Two of the datasets are publicly available. In addition, we have also created two new annotated datasets.

Stanford Twitter Sentiment (STS). This dataset has been commonly used in the literature for evaluation. It comprises of a training and testing set. The training set consists of 800,000 positive and 800,000 negative tweets classified using noisy labels. The testing set consists of 177 negative and 182 positive tweets which are manually classified. The query used to retrieve the tweet is also given for each tweet and is treated as the target or the aspect in our experiment. Some of the targets include *kindle2*, *aig*, *jquery*, *booz allen*, and *obama*.

Sanders Twitter Corpus (STC)[1]. This is a set of 5513 manually classified tweets. Each tweet is classified with respect to four different targets (*apple*, *google*, *microsoft* and *twitter*). The tweets are classified to be positive, negative, neutral or irrelevant. Due to Twitter's Terms of Service, the author only provided a script to download the tweets instead of the tweets themselves. We only manage to download 4970 of them as many of the accounts associated with the tweets have been set to be private. Ignoring the neutral and irrelevant tweets, we managed to obtain 511 positive and 561 negative tweets used for our experiments.

Telecommunication Company Dataset 1 & 2 (TCD1/TCD2). We have also created two new annotated datasets based on tweets from two different telecommunication companies in Singapore. The tweets were first retrieved using the Twitter search API over the period of two months from December 2012. We then use a subjectivity classifier (similar to the tweet-level sentiment classifier described in section V) to pick out tweets which are likely to be positive or negative. Each tweet is annotated with a list aspects that is relevant to the company, the associated sentiment word, and the polarity of the aspect (e.g. *[3G, annoying, -]*, *[LTE, cool, +]*). Sentiment words which cannot be associated with any explicit aspect are annotated with the *general* aspect. In our experiments, we have chosen to ignore the *general* aspect. Finally, we obtain 545 (758) reviews and 459 (566) classification cases in TCD1 (TCD2).

B. Experiments

Table III shows the sentiment classification accuracy on the STS and STC datasets by using the STS training data to build the classifier. MaxEnt (NB) is the tweet-level classifier described in section V using Maximum Entropy (Naïve Bayes). The target of the tweet (provided by the STS/STC) is used as the only expected aspect. Aspects which are not the expected aspect are ignored. Layered is the layered classification approach described in section V without considering aspect-ranking (subsection III.B) and sentiment expressions (section IV). Ngram considers sentiment

[1] Downloaded from http://www.sananalytics.com/lab/twitter-sentiment/

expressions (section IV) and performs aspect ranking and selection. It is possible that people comment on various aspects of the target without directly mentioning the target itself, thus these related aspects should also be treated as the expected aspects. For example, some of the related aspects of the target *jquery* might be *plugin* and *ajax*. These related aspects are obtained by performing the aspect-sentiment extraction and ranking method (subsections III. A and III.B) on tweets containing the target from the STS training dataset. In addition, during the testing phase, we also perform co-reference resolution on the tweets to further expand the set of expected aspects by considering nouns which are the same referent as the given target. The final outcome of aspect ranking and selection is similar to extended targets in [10]. As the STS training dataset does not include the targets and its associated polarity, we are not able to use this dataset to incorporate the target-dependent features (the second and third method from subsection V.A) in the training and testing. Nevertheless, this is investigated when we are using the STC as training data (Table V).

Table IV shows the performance of our aspect-based classifier performance on the STS and STC dataset. In order to classify a tweet to be positive or negative, we employ the same approach as described in the layered classification (subsection V. A) without a tweet-level classifier. As there might not be a classification result, we measure the performance by the precision, recall, and f-measure metrics.

Table V shows the sentiment classification accuracy for STS when using the STC as the training data. The STC training dataset allows us to evaluate the three different approaches of incorporating the aspect-based sentiment classifier results into the tweet-level classifier. MaxEnt+Target follows the second approach described in subsection V.A when a maximum entropy classifier is used and MaxEnt+Target (backoff) follows the third approach described in subsection V.A. We compare our approach to [10] which incorporates target-dependent features from a dependency parser as well as target-independent features. They adopt a more elaborate definition of target-dependent features where they differentiate between how a token is associated with the target.

Tables VI and VII show the performance of the aspect-based classifier (Aspect) on TCD1 and TCD2 respectively. Unlike the experiments for STS and STC, in this experiment, the classifiers are required to produce a polarity for each aspect found in the tweet. We investigate the case where the list of aspects is known and given to the classifier, and the case where the list of aspects is unknown and has to be determined by the classifier. For the unknown aspect case, aspect ranking and selection is performed using the method described in subsection III.B on the entire TCD1/TCD2 dataset. We also define a lexicon approach as baseline. This approach would scan the left and right of a given aspect candidate to find the first token that is contained in the sentiment lexicon/swear word list. For example, assume that *3G* is an aspect candidate in the tweet *"the slow 3G really sucks"*, after scanning to the left and right of the aspect candidate, *[3G, slow, -]* and *[3G, sucks, -]* are generated, and added to the result list. Furthermore, we compare our approach with a product review-based aspect-sentiment classifier [12].

TABLE III. SENTIMENT CLASSIFICATION ACCURACY ON STS (TRAINING ON STS TRAINING DATASET)

Method	STS	STC
Go et al. [8]	83	-
Speriosu et al. [21]	84.7	-
Saif et al. [20]	86.3	-
MaxEnt	84.7	77.7
MaxEnt+Layered	87.7	**80.2**
MaxEnt+Layered+Ngram	**88.3**	**80.2**
NB	84.1	75.1
NB+Layered	86.3	79.1
NB+Layered+Ngram	87.2	79.3

TABLE IV. SENTIMENT CLASSIFICATION ACCURACY OF ASPECT-BASED SENTIMENT CLASSIFIER (IN TERMS OF PRECISION, RECALL, F-MEASURE)

Dataset	Method	P	R	F
STS	Aspect	93.1	40.4	56.3
	Aspect+Ngram	**93.7**	**44.0**	**59.9**
STC	Aspect	87.4	38.5	53.5
	Aspect +Ngram	**87.5**	**46.8**	**61.0**

TABLE V. SENTIMENT CLASSIFICATION RESULTS ON STS (USING THE WHOLE STC DATASET FOR TRAINING)

Method	Accuracy (%)
Jiang et al. [10][2]	82.7
MaxEnt	81.8
MaxEnt+Layered	**84.1**
MaxEnt+Layered+Ngram	**84.1**
MaxEnt+Target	82.1
MaxEnt+Target (backoff)	82.7

TABLE VI. SENTIMENT CLASSIFICATION RESULTS ON TCD1 (IN TERMS OF PRECISION, RECALL, F-MEASURE)

Method		P	R	F
Known Aspect	Aspect	**83.6**	75.4	79.3
	Aspect+Ngram	83.3	76.9	**80.0**
	Lexicon	73.2	**80.8**	76.8
	Lek and Poo [12]	75.2	25.1	37.6
Unknown Aspect	Aspect	68.3	70.4	69.3
	Aspect +Ngram	68.3	71.5	**69.8**
	Lexicon	57.9	**76.9**	66.0
	Lek and Poo [12]	**70.6**	20.9	32.3

TABLE VII. SENTIMENT CLASSIFICATION RESULTS ON TCD2 (IN TERMS OF PRECISION, RECALL, F-MEASURE)

Method		P	R	F
Known Aspect	Aspect	**88.4**	76.5	82.0
	Aspect+Ngram	88.2	77.9	**82.7**
	Lexicon	81.5	**82.7**	82.1
	Lek and Poo [12]	69.1	26.5	38.3
Unknown Aspect	Aspect	68.8	67.8	68.3
	Aspect+Ngram	**68.9**	68.9	**68.9**
	Lexicon	54.7	**75.4**	63.4
	Lek and Poo [12]	56.2	22.1	31.8

[2] This accuracy can be further improved to 85.2% if we perform +Layered+Ngram

C. Discussion

The experimental results show that MaxEnt consistently outperforms NB (Table III). Our tweet-level sentiment classifiers (MaxEnt and NB) rival the state-of-the-art approaches [8, 21] on the STS dataset. Based on the improvement over Go et al. [8] which is using unigram and bigram features, we affirm that lexical features constructed from a sentiment lexicon is useful for improving the performance of Twitter sentiment classification. By incorporating the aspect-based classifier using a layered classification approach, we were able to outperform the best reported system [20] in the literature.

Saif et al. [20] use a third-party entity extraction service to map entities into a concept. For example, "*iPhone*" and "*Macbook*" is mapped to the "*Product/Apple*" concept, and based on the idea that Apple products tend to have a positive polarity, this introduces an additional prior polarity bias into the classifier. This approach works well if the general polarity of the concept based on the training and testing tweet does not vary much. However, if we are classifying a tweet which expresses an opinion which is out of the norm, this approach might wrongly introduce a prior polarity bias. Our approach on the other hand does not take into account this form of trend-based information.

We see that a layered classification approach to incorporate the aspect-based classifier can consistently boost the performance of an existing classifier (Tables III and V). This is mainly due to the high precision of the aspect-based classifier (Table IV). The error cases of our aspect-based classifier can be mainly summarized into four major types.

First, since we are processing the tweets on a sentence-level, it does not work well for cases where there are inter-sentence dependencies. For example the following tweet on Apple: "*Dear @Apple, it's me again. Thank for beautiful new iOS features. But I miss some of the old ones. Lk making calls & texts*" is annotated as negative. However, since we process the tweets on a sentence-level, "*Thank for beautiful new iOS features*" would result *[iOS, beautiful, +]* and because there is no other aspect detected from other sentences in the tweet, this tweet is erroneously classified as positive. This constitutes about 20 to 30% of the error.

Second, it is possible that sentiment words are linked to a wrong target, or are wrongly associated with the target. For example, consider a tweet talking about Google, *"Sorry #Apple #Google and #Samsung just made you look bad. #Android is king"*, our system generates this tuple *[google, sorry, -]* causing the tweet to be wrongly classified as negative. This type of error constitutes about 10 to 30% of the error.

Third, it is possible that the sentiment lexicon wrongly classify a sentiment word since it does not consider the context. For example, the following tweet from STS "*@Lou911 Lebron is MURDERING shit*" is classified as positive. However, both "*MURDERING*" and "*shit*" are negative words. Without having background context, it is difficult (even for a human) to classify the tweet accurately. Some words can also take on different polarity depending on how they are used. Nevertheless, for some situations, it is possible that the multi-word sentiment expressions (generated from section IV) can help to correct the polarity. This type of error constitutes 15 to 20% of the error.

Lastly, our approach does not work well for sarcastic tweets. For example, consider this tweet *"@apple thank you for ruining my 3GS with #iOS5."*, *Apple* is associated with *thanks* and wrongly classified as positive. The precision of STC is much lower compared to STS because there are a lot more sarcastic tweets on STC compared to STS. 26 out of the 1072 tweets on STC are sarcastic but there are just 2 sarcastic tweets on STS. This type of error constitutes 10 to 15% of the error.

The low recall of our aspect-based classifier is due to the fact that many of the opinions in the tweet are not associated with any aspect. Thus by looking at just the aspect-sentiment extraction, we cannot determine the polarity of the tweet. Instead, it is better to adopt a text classification approach which considers the distribution of tokens for classification.

We also note that there are some issues with the STS testing set. Even though the query for retrieving a tweet is provided for every tweet, each tweet is manually classified based on the overall sentiment rather than with respect to the query (which we use as the expected target). For example, a tweet on *surgery* "*Recovering from surgery..wishing @julesrenner was here :(*" is classified as positive by our classifier because of the tuple *[surgery, recovering, +]*, but this is manually annotated as negative because of "*wishing @julesrenner was here*". Furthermore, some of the queries in STS are also not noun such as "*itchy*", "*eating*", and "*driving*" so they cannot be treated as the target.

From table V, we see that the layered classification approach produces the most improvement to an existing classifier compared to incorporating target-dependent features [10]. The latter suffers from the problem of sparsity of target-dependent features and can only work well if there are large amount of training examples. However, large amount of training data that is annotated with the target along with its associated polarity is difficult to obtain and too tedious to annotate, thus making this approach not scalable. We also did a 10-fold cross validation on STC by dividing the STC dataset into two, using half for training and half for testing. After incorporating target-dependent features, there is no improvement over the tweet-level classifier's accuracy of 83.6%. However, by using the layered classification approach, we managed to boost this accuracy to 84.2%

On TCD1 and TCD2 datasets, we observe that our aspect-based classifier has better precision and better overall F-measure compared to the baseline method. The lexicon method has better recall but suffers from low precision. By incorporating sentiment expressions, the recall of the classifier is improved without much penalty on the precision. The approach by [12] which uses a dependency parser and extraction rules to obtain the aspect-sentiment suffers from low recall. This further demonstrates that existing linguistic tools do not work well on tweet data due to the informal characteristics of tweets. Nevertheless, the improvement of our aspect-based classifier over the lexicon method shows that POS information is still useful.

VII. Conclusion and Future Work

Previous work in Twitter sentiment classification has performed tweet-level sentiment classification. In this paper, we present an approach to perform aspect-based sentiment classification for Twitter. Our aspect-based sentiment classifier makes use of a POS tagger, a sentiment lexicon and a few gazetteer lists to produce results of the form *[aspect, sentiment words, polarity]*. We also describe a method to mine sentiment expressions and show that these sentiment expressions can further improve the classification performance. We experimented various ways to incorporate the results from the aspect-based classifier into conventional tweet-level classifier. The experimental results suggest that a layered classification approach which uses the aspect-based classifier as the first layer classification and the tweet-level classifier as the second layer classification is more effective than a classifier trained using target-dependent features. This approach is able to consistently improve the performance of existing sentiment classifiers.

In the future, we plan to consider inter-sentence and inter-aspect/sentiment pair relationship. We also plan to consider contextual information when classifying sentiment words. In addition, we would explore using a separate classifier to detect sarcastic tweets.

Acknowledgement

This work is supported by Singapore Ministry of Education Academic Research Fund Tier 1 under the grant number: T1251RES1002.

References

[1] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in *Proceedings of the Conference on International Language Resources and Evaluation*, Valletta, Malta, 2010.

[2] L. Barbosa and J. Feng, "Robust Sentiment Detection on Twitter from Biased and Noisy Data," in *Proceedings of the International Conference on Computational Linguistics: Posters*, Beijing, China, 2010, pp. 36–44.

[3] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *J. Comput. Science, Vol. 2, No. 1*, pp. 1–8, 2011.

[4] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A. P. Sheth, "Extracting Diverse Sentiment Expressions with Target-Dependent Polarity from Twitter," in *Proceedings of the International Conference on Weblogs and Social Media*, Dublin, Ireland, 2012.

[5] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using Twitter hashtags and smileys," in *Proceedings of the International Conference on Computational Linguistics: Posters*, Beijing, China, 2010, pp. 241–249.

[6] B. J. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points.," *Science, Vol. 315*, Jan. 2007.

[7] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for Twitter: annotation, features, and experiments," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies: Short Papers – Vol. 2*, Portland, Oregon, 2011, pp. 42–47.

[8] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," unpublished, 2009.

[9] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Seattle*, Washington, 2004, pp. 168–177.

[10] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies – Vol. 1*, Portland, Oregon, 2011, pp. 151–160.

[11] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters," *Computational Intelligence, Vol. 22, No. 2, Special Issue on Sentiment Analysis*, pp. 110–125, 2006.

[12] H. H. Lek and D. C. C. Poo, "Sentix: An Aspect and Domain Sensitive Sentiment Lexicon," in *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, Athens, Greece, 2012, pp. 261 – 268.

[13] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the Web," in *Proceedings of the International Conference on World Wide Web*, Chiba, Japan, 2005, pp. 342–351.

[14] H. Mao, S. Counts, and J. Bollen, "Predicting Financial Markets: Comparing Survey,News, Twitter and Search Engine Data," *CoRR, vol. abs/1112.1051*, 2011.

[15] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing-Vol. 10*, 2002, pp. 79–86.

[17] A.-M. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," in *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, pp. 339–346.

[18] G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding Domain Sentiment Lexicon through Double Propagation," in *International Joint Conference on Artificial Intelligence*, 2009, pp. 1199–1204.

[19] J. Read, "Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification," in *Proceedings of the ACL Student Research Workshop*, Ann Arbor, Michigan, 2005, pp. 43–48.

[20] H. Saif, Y. He, and H. Alani, "Alleviating Data Sparsity for Twitter Sentiment Analysis," in *Workshop of Making Sense of Microposts co-located with WWW 2012*, Lyon, France, 2012, pp. 2–9.

[21] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph," in *Proceedings of the Workshop on Unsupervised Learning in NLP*, Edinburgh, Scotland, 2011, pp. 53–63.

[22] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," in *Proceedings of the International Conference on Weblogs and Social Media*, Washington, DC, USA, 2010.

[23] P. D. Turney and M. L. Littman, "Measuring Praise and Criticism: Inference of Semantic Orientation from Association," *ACM Transactions on Information Systems, Vol. 21, No. 4*, pp. 315–346, 2003.

[24] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle," in *Proceedings of the ACL System Demonstrations*, Jeju Island, Korea, 2012, pp. 115–120.

[25] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, pp. 347–354.

[26] L. Zhang and B. Liu, "Identifying noun product features that imply opinions," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies: Short Papers – Vol. 2*, Stroudsburg, PA, USA, 2011, pp. 575–580.