Sentiment Lexicon Interpolation and Polarity Estimation of Objective and Out-Of-Vocabulary Words to Improve Sentiment Classification on Microblogging

Yongyos Kaewpitakkun, Kiyoaki Shirai, Masnizah Mohd
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi City, Ishikawa, Japan 923-1292
Email: {s1320203,kshirai,masnizah}@jaist.ac.jp

Abstract

Sentiment analysis has become an important classification task because a large amount of user-generated content is published over the Internet. Sentiment lexicons have been used successfully to classify the sentiment of user review datasets. More recently, microblogging services such as Twitter have become a popular data source in the domain of sentiment analysis. However, analyzing sentiments on tweets is still difficult because tweets are very short and contain slang, informal expressions, emoticons, mistyping and many words not found in a dictionary. In addition, more than 90 percent of the words in public sentiment lexicons, such as SentiWordNet, are objective words, which are often considered less important in a classification module. In this paper, we introduce a hybrid approach that incorporates sentiment lexicons into a machine learning approach to improve sentiment classification in tweets. We automatically construct an *Add-on lexicon* that compiles the polarity scores of objective words and out-of-vocabulary (OOV) words from tweet corpora. We also introduce a novel feature weighting method by interpolating sentiment lexicon score into uni-gram vectors in the Support Vector Machine (SVM). Results of our experiment show that our method is effective and significantly improves the sentiment classification accuracy compared to a baseline uni-gram model.

1 Introduction

Sentiment analysis and opinion mining is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes and emotions from written language (Liu, 2010). Recently, Twitter has become an important resource for sentiment analysis. People express their opinions and feelings using Twitter and these data can be grabbed publicly through Twitter API. There are two main approaches to sentiment analysis: lexicon-based and machine learning-based techniques. Several researchers have combined these two techniques (Kumar et al., 2012; Mudinas et al., 2012; Saif et al., 2012; Fang et al., 2011; Hung et al., 2013). This study adopts a similar approach; we seek to combine the prior polarity knowledge from the lexicon-based method and the powerful classification algorithm from the machine learning-based method. Two main motivations of this approach are discussed below.

The initial motivation is to revise the polarity of objective and out-of-vocabulary words in the public sentiment lexicon to improve Twitter sentiment classification. In the lexicon-based approach, sentiment classification is done by comparing the group of positive and negative words looked up from the public lexicon. For example, if the document contains more positive words than negative words, it will be classified as positive. Several public lexical resources such as ANEW[1], OpinionFinder[2], SentiStrength[3], SentiWordNet[4] and SenticNet[5] lexicon are available for this type of analysis. SentiWordNet or "SWN" (Esuli et al., 2010) has become one of the most fa-

[1] http://neuro.imm.dtu.dk/wiki/A_new_ANEW/

[2] http://mpqa.cs.pitt.edu/opinionfinder/

[3] http://sentistrength.wlv.ac.uk/

[4] http://sentiwordnet.isti.cnr.it/

[5] http://sentic.net/

mous and widely used sentiment lexicons because of its huge vocabulary coverage. SentiWordNet is an extended version of WordNet[6], where words and synsets in WordNet are augmented with their sentiment score. SWN 3.0 contains more than 100,000 synsets. However, more than 90% of these are classified as objective words (Hung et al., 2013); which are usually considered less important in the classification process. Furthermore, lexicon-based sentiment analysis over Twitter faces several challenges due to the short informal language used. Tweets are usually short and contain lots of slang, emoticons, abbreviations or mistyped words. Most of them are not contained in the public lexicon, which are called out-of-vocabulary (OOV) words. Both objective and OOV words may have implicit sentiment, especially in some specific domains or group of users; thus, it could be better to modify an existing public sentiment lexicon, such as SentiWordNet, by incorporating the polarity of objective and OOV words. One possible way to revise SentiWordNet is to estimate the polarity scores of sentiment unknown words based on the polarity of the sentences including them in the corpus. For example, let us suppose that the objective word "birthday" appears many more times in positive tweets than in objective or negative tweets. This word could be revised as a positive word in the sentiment lexicon. On the other hand, when the OOV word "ugh" appears many more times in negative tweets than in objective or positive tweets, it could be newly classified as a negative word. In this work, we aim to build an add-on lexicon covering the estimated polarity scores for both objective words and OOV words in the SentiWordNet.

The secondary motivation is to incorporate the prior polarity knowledge from the sentiment lexicon into powerful machine learning classifier, such as the Support Vector Machine (SVM), as extra information. Among many machine learning techniques, SVM has achieved the great performance in the sentiment classification task. The uni-gram feature has been widely and successfully used in sentiment analysis, especially in user review datasets. Since tweets are much shorter than user reviews, however, the use of only the uni-gram feature may cause a data sparseness problem. One possible way to solve this problem is to integrate the information from the sentiment lexicon to supervised algorithms as extra knowledge. Recently, some researchers incorporate information derived from a lexicon into machine learning by augmenting sentiment lexicon as extra polarity group feature to uni-gram (O'Keefe et al., 2009) or simply replacing uni-gram with a lexicon score (Hung et al., 2013). In this work, we present an alternative way to incorporate lexical information into a machine learning algorithm by interpolating a score in the sentiment lexicon into a score of uni-gram feature in vector weighting. Our experiment results show that the proposed lexicon interpolation weighting method with revised polarity estimation of objective and OOV words is effective and significantly improves the sentiment classification accuracy compared to the baseline uni-gram model.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes our proposed method and framework including data pre-processing, polarity estimation technique and sentiment lexicon incorporation and feature weighting method. Section 4 describes results of the experiments and discussion. Finally, conclusions and direction for future work are discussed in Section 5.

2 Related Work

Early work on Twitter sentiment analysis used two approaches in traditional sentiment analysis on normal texts: machine learning-based and lexicon-based approaches. Recently, some studies have combined these two approaches and achieved relatively better performance in two ways. The first is to develop two classifiers based on these two approaches separately and then integrate them into one system. The second is to incorporate lexicon information directly into a machine learning classification algorithm. In the first way, Kumar et al. (2012) used a machine learning-based method to find the semantic orientation of adjectives and used a lexicon-based method to find the semantic orientation of verbs and adverbs. The overall tweet sentiment is then calculated using a linear interpolation of the results from both methods. Mudinas et al. (2012) presents concept-level sentiment analy-

[6] http://wordnet.princeton.edu/

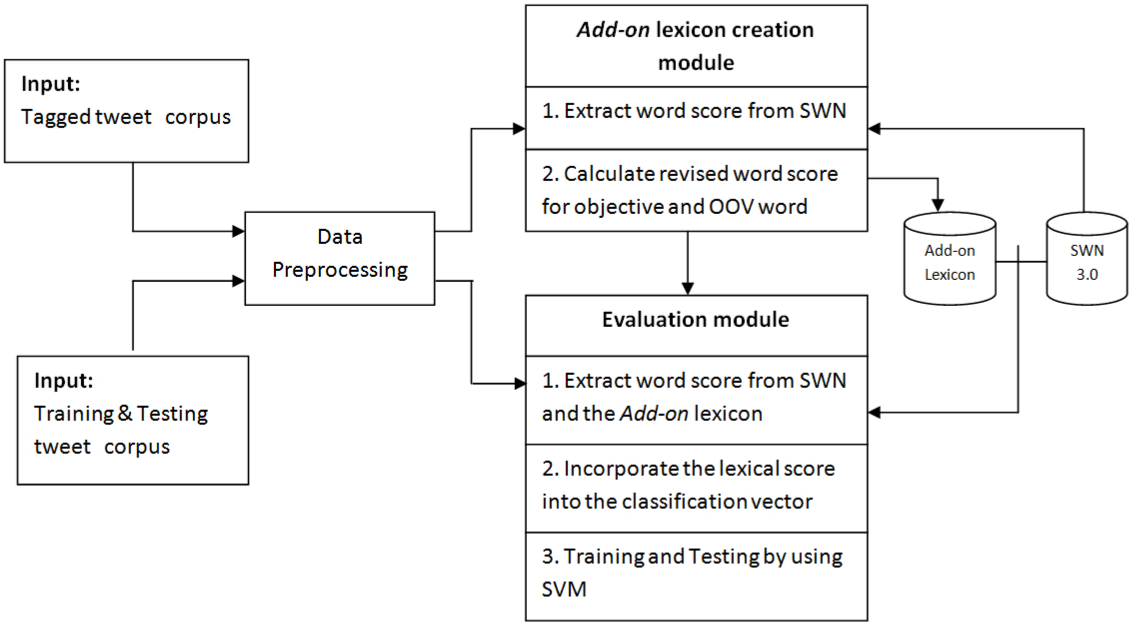

Input:
Tagged tweet corpus
Input:
Training & Testing tweet corpus
Data Preprocessing
Add-on lexicon creation module
1. Extract word score from SWN
2. Calculate revised word score for objective and OOV word
Evaluation module
1. Extract word score from SWN and the *Add-on* lexicon
2. Incorporate the lexical score into the classification vector
3. Training and Testing by using SVM
Add-on Lexicon
SWN 3.0

Figure 1: System framework.

sis system, which are called pSenti. Their system used a lexicon for detecting the sentiment of words and used these sentiment words as features in the machine learning-based method. Results from both lexicon and machine learning were combined together to calculate the final overall sentiment scoring. In the second way, Saif et al. (2012) utilized knowledge of not only words but also semantic concepts obtained from a lexicon as features to train a Naive Bayes classifier. Fang et al. (2011) automatically generated domain-specific sentiment lexicon and incorporated it into the SVM classifier. They applied this method for identifying sentiment classification in a product reviews. Recently, Hung et al. (2013) reported that more than 90 percent of words in SentiWordNet are objective words that are often considered useless in sentiment classification. So, they reassigned proper sentiment values and tendency of such objective words in a movie review corpus and incorporated these sentiment scores into the machine learning-based method. In this paper, we reevaluate the sentiment score of not only objective words but also out-of-vocabulary (OOV) words; which are common in tweets due to informal message used. We also propose an alternative way to incorporate the sentiment lexicon knowledge into the machine learning algorithm. We will propose sentiment interpolation weighting method that interpolates lexicon scores into uni-gram scores in the vector representation of the SVM classifier. Our method is described in detail in the next section.

3 Approach

Our two-step hybrid sentiment analysis system has been developed by combining lexicon-based and machine learning-based approaches. In the first step, the add-on lexicon has been created by reevaluating the polarity scores of objective words and out-of-vocabulary (OOV) words extracted from a specific tweet corpus. After that, the score from both the public lexicon and add-on lexicon will be incorporated into a feature vector as extra prior knowledge in four different ways that will be described in Subsection 3.3. The main advantage of our approach is the extra sentiment polarity information from both the public and add-on lexicon will be incorporated to the powerful machine learning algorithm. It can help the supervised learned classifier to identify the sentiment of tweets more precisely, even when tweets contain words that are not found in the public lexicon or less frequently appeared in the training set. The overall system framework is shown in Figure 1.

3.1 Data preprocessing

The data preprocessing process consists of part-of-speech tagging, lemmatizing, and stop word and URL removal. In the first step, tweets are POS tagged by the TweetNLP POS Tagger[7], which is trained specially from Twitter data. After that, all words are lemmatized by the Stanford lemmatizer[8]. We also reduce the number of letters that are repeated more than two times, i.e. "heellllooooo" is replaced by "heelloo". Finally, the common stop words and URL are removed because they represent neither sentiment nor semantic concept.

3.2 Add-on lexicon creation

As discussed above, SentiWordNet has become a famous and useful lexicon for sentiment analysis due to its broad coverage; however more than 90 percent of words in SentiWordNet are objective words. Moreover, lots of words in tweets are slang, informal or mistyped words that are not included in the lexicon. Based on this observation, we aim to build an add-on lexicon by compiling both objective and OOV words with their newly estimated sentiment score. Word scores are estimated based on the assumption that the polarities of words are coincident with the polarity of their associated sentences, which seems reasonable due to the short length of tweet messages. In other words, if the word frequently appears in the positive (or negative) tweets, its polarity might be positive (or negative).

In the creation of the add-on lexicon, the sentiment score of a word is calculated based on the probability that the word appears in positive or negative sentences in a sentiment tagged corpus. There are two steps. In the first step, the words from preprocessing step are extracted with their score in SentiWordNet by using Equation (1). As we will describe in Subsection 3.3, this score is used as the weight of the feature vector. In the add-on lexicon creation, SentiWordNet is just used to check if the word is an objective word ($SWNScore(w_i) = 0$) or OOV word, then objective and OOV words will be sent to the revised polarity estimation step. The revised scores for these words are calculated by Equation (2).

$$SWNScore(W_i) = SWNScore_{POS}(w_i) - SWNScore_{NEG}(w_i) \tag{1}$$

$$Score(w_i) = \begin{cases} Score_{POS}(w_i), \\ \quad \text{if } Score_{POS}(w_i) > Score_{NEG}(w_i). \\ (-1) \times Score_{NEG}(w_i), \\ \quad \text{if } Score_{POS}(w_i) < Score_{NEG}(w_i). \end{cases} \tag{2}$$

where,

$$Score_{POS}(w_i) = \frac{P(positive|w_i)}{P(positive)}$$

$$Score_{NEG}(w_i) = \frac{P(negative|w_i)}{P(negative)}$$

$$P(positive|w_i) = \frac{No.\ of\ w_i\ in\ positive\ tweets}{No.\ of\ w_i\ in\ dataset}$$

$$P(negative|w_i) = \frac{No.\ of w_i\ in\ negative\ tweets}{No.\ of\ w_i\ in\ dataset}$$

$$P(postitive) = \frac{No.\ of\ positive\ tweets}{No.\ of\ all\ tweets}$$

$$P(negative) = \frac{No.\ of\ negative\ tweets}{No.\ of\ all\ tweets}$$

In the second step, since scores in SentiWordNet are in the range of -1 to 1, we have to convert the revised word scores into the same interval. In this case, we use a Bipolar sigmoid function (Fausett, 1994) because it is continuous and returns a value from -1 to 1. The conversion formula is shown in Equation (3).

$$Score(w_i)^{'} = sigmoid(Score(w_i)) \tag{3}$$

where, $sigmoid(x) = \frac{2}{(1+e^{-x})} - 1$

The revised polarity score may be unreliable if the frequency of the word is too low, or the difference between positive and negative tendency is not great enough. Therefore, two thresholds are introduced. Threshold 1 (T1) is the minimum number of words in the dataset and threshold 2 (T2) is the minimum difference between positive and negative word orientation scores ($Score_{POS}(w_i)$ and $Score_{NEG}(w_i)$). The objective and OOV words with their scores are added to the add-on lexicon only when equation (4) is fulfilled.

$$\begin{gathered} Frequency\ of\ w_i\ in\ dataset \geq T_1 \\ |Score_{POS}(w_i) - Score_{NEG}(w_i)| \geq T_2 \end{gathered} \tag{4}$$

3.3 Lexicon score incorporation and feature weighting methods

In this subsection, the word scores from both SentiWordNet and the add-on lexicon will be incor-

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

[8] http://nlp.stanford.edu/software/

porated into the SVM classification features as extra prior information in four different ways: sentiment weighting, sentiment augmentation, sentiment interpolation and sentiment interpolation plus. We start with the baseline uni-gram features, followed by our proposed sentiment lexicon incorporation method. Note that we ignore word sense disambiguation problem although the sentiment score is associated not with a word but with a synset in SWN. When SWN is consulted to obtain a sentiment score for a polysemous word, the first word sense in SWN is always chosen because it is the most representative sense of each word.

3.3.1 Uni-gram and POS Features

Uni-gram and POS features are common and widely used in the domain of sentiment analysis. There are many feature weighting schemes for the uni-gram. In this work, we use the combination of uni-gram and POS features with term presence weighting as the baseline method. As a result, the weight value of words(POSs) is 1 if they are present, otherwise 0.

3.3.2 Sentiment Weighting Features

In this method, the feature weights of uni-gram binary vectors will be simply replaced with the word sentiment scores (Equation (1) or (3)) from the lexicon. Note that the weight is set to 0 if the word does not appear in the tweet.

3.3.3 Sentiment Augmentation Features

In this method, words will be classified into 3 groups: positive, objective and negative, based on their scores in the lexicon. Then, these sentiment group features are augmented to the original uni-gram vector. There are three additional features that are the percentage of positive, objective and negative words in a tweet, where the sum of the weights of these three features would be equal to one.

3.3.4 Sentiment Interpolation Features

In this method, we proposed a new incorporation method where the word score from the lexicon will be interpolated into the original uni-gram feature weight. The weight of the new interpolated vector is shown in Equation (5). Note that uni-gram score is always 1 in our model.

Table 1: Summary of feature and weighting methods.

Methods	**Feature weight value**	**Additional features**
Uni-gram + POS	1	No
Sentiment Weighting	Lexicon score	No
Sentiment Augmentation	1	percentage of positive, objective and negative word in a tweet
Sentiment Interpolation	Equation (5)	No
Sentiment Interpolation Plus	Equation (5)	percentage of positive, objective and negative word in a tweet

$$Weight = \alpha \; Uni\text{-}gram \; score + (1-\alpha) \; Lexicon \; score \qquad (5)$$

The parameter α $(0 \leq \alpha \leq 1)$ is used for controlling the influence between the uni-gram model and the sentiment lexicon model. When α is equal to 1, the weight is the fully uni-gram model, and when α is 0, the weight is the fully sentiment weighting model.

3.3.5 Sentiment Interpolation Plus Features

In this method, we combine sentiment interpolation and sentiment augmentation together. Therefore, three additional augmentation features (Subsection 3.3.3) will be added to the sentiment interpolation vector (Subsection 3.3.4) as the extra features.

The summary of all features and weight values are shown in Table 1. Please note that the weight of the feature is always 0 if it does not appear in the tweets.

4 Evaluation

In this section, we present the results of two experiments. The first experiment was conducted with Positive-Neutral-Negative classification over full datasets (3-way classification). In the second experiment, we discarded neutral tweets and conducted the experiment with Positive-Negative classification over datasets of only positive and negative tweets. The detailed results are shown in Section 4.3. In addition, we used LIBLINEAR[9] developed by Fan et al. (2008) with default setting for training the SVM classifier.

[9] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

Table 2: Sanders corpus.

Subset	Used for	# Pos	# Neu	# Neg	# Total
1	Add-on lexicon creation, Training	319	1,319	345	1,983
2	Testing	109	455	114	678

Table 3: SemEval 2013 corpus.

Subset	Used for	# Pos	# Neu	# Neg	# Total
0	Development	1,297	1,401	475	3,173
1	Add-on lexicon creation, Training	2,272	3,083	884	6,239
2	Testing	372	441	187	1,000

4.1 Data set

4.1.1 Sanders Dataset

The Sanders corpus[10] consists of 5,512 tweets on four different topics (Apple, Google, Microsoft, and Twitter). Each tweet was manually labeled as positive, negative, neutral or irrelevant. After removing irrelevant and duplicate tweets, 2,661 tweets remained. Then, the dataset was randomly divided into two subsets. The first sub-dataset was used for the add-on lexicon creation part and training part, while the second was used for the testing (evaluation) part. Detailed information on this corpus is shown in Table 2. We used the Sanders dataset as a representative of small and domain-specific corpus.

4.1.2 SemEval 2013 Dataset

The SemEval 2013 corpus (Nakov et al., 2013) consists of about 15,000 tweets that were created for Twitter sentiment analysis (task 2) in the Semantic Evaluation of Systems Challenge 2013. Each tweet was manually labeled as positive, negative or neutral by Amazon Mechanical Turk workers. This dataset consists of a variety of topics. Among the full dataset, only 10,534 tweets could be downloaded, because some of them were protected or deleted. This dataset was also randomly divided into three subsets. Detailed information on this corpus is shown in Table 3. Note that the development set was used for parameter tuning. We used the SemEval 2013 dataset as a representative of a large and general corpus.

In addition, the percentages of objective words and OOV words after data preprocessing in both corpora are shown in Table 4.

4.2 Parameter optimization

As described in Subsection 3.2, in the add-on lexicon creation process, two thresholds can play an important role to control the number of revised polarity words. The objective and OOV words should not be revised if their estimated scores are not reliable enough. To investigate an optimal value for the threshold T1, we conducted a sensitivity test on the SemEval 2013 development dataset (subset 0 in Table 4). Note that the threshold T2 was set to 0.2 by the preliminary experiment. Figures 2 a and b show the accuracy of our method for various values of T1 using interpolation plus weighting method in a 3-way and a positive-negative classification, respectively. In these graphs, the horizontal axis indicates the ratio of the number of words in the add-on lexicon to that of the corpus. The results show that, in 3-way classification, the classifier achieved better performance when the numbers of revised polarity words were smaller than the case of positive-negative classification. The accuracy reached its peak with the percentage of revised polarity words set around 0.5% (in 3-way classification) and 1.2% (in positive-negative classification). We did not investigate the optimum for the threshold T1 in the Sanders corpus due to the insufficient number of tweets, but set T1 so that the percentage of the number of the add-on lexicon is the same as in the optimized value in the SemEval 2013 dataset. Based on this observation, two thresholds were set as shown in Table 5.

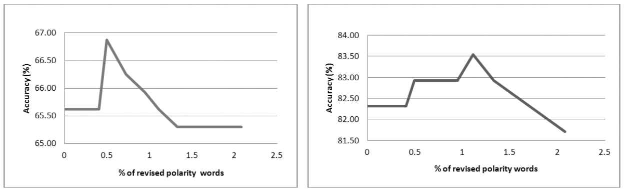
Table 4: Percentages of objective and OOV words in the two corpora.

Corpus	Objective words	OOV words
Sanders	26.61%	57.73%
SemEval 2013	24.01%	66.55%

4.3 Results

Table 6 and 7 show the results of the 3-way and positive-negative classification, respectively. They reveal the average of precision, recall and F1-

[10] http://www.sananalytics.com/lab/twitter-sentiment/



(a) positive-neutral-negative corpus

(b) positive-negative corpus

Figure 2: The classification accuracy vs. number of revised polarity words on the development dataset.

Table 5: Threshold parameter setting based on % of revised words.

Corpus	Task	T1	T2	Vocab. size	*1	*2
Sanders	3-way	45	0.20	5,145	24	0.46%
	pos-neg	25	0.20	5,145	60	1.17%
SemEval 2013	3-way	60	0.20	15,366	78	0.50%
	pos-neg	35	0.20	15,366	173	1.12%

*1 = No. of revised words, *2 = % of revised words

measure over positive and negative classes as well as accuracy (Acc) for both Sanders and SemEval 2013 datasets. Five methods (including the baseline) described in Subsection 3.3 with and without the add-on lexicon are compared. In the experiment, the coefficient α in Equation (5) was initially set to 0.5 for maintaining the balance of uni-gram and lexicon score. The sensitivity of α will be investigated in Subsection 4.6.

4.4 Effect of the add-on lexicon

In this section, we compare the performance of the add-on lexicon to the original SentiWordNet lexicon. Figure 3 shows the accuracy (the average of both 3-way and positive-negative classification tasks and both datasets) of the models with original SWN and SWN plus the add-on lexicon using four different feature weighting methods. It indicates that the add-on lexicon significantly improved the accuracy in the sentiment weighting and slightly improved the accuracy in the sentiment interpolation and sentiment interpolation plus. In the case of sentiment augmentation, the accuracies were almost the same. In addition, the combination of sentiment interpolation plus the add-on lexicon achieved the highest accuracy.

Table 8: Average accuracy improvement when using SWN vs. SWN plus the add-on lexicon in 3-way and positive-negative classification.

Classification	Sentiment Interpolation	Sentiment Interpolation Plus
3-Way	+0.27%	+0.25%
Positive-Negative	+2.42%	+2.06%

When the add-on lexicon was applied, the performance improved more in positive-negative classification than in positive-neutral-negative (3-way) classification. Table 8 shows the average of both datasets of accuracy improvement in 3-way and positive-negative classification with and without the add-on lexicon when using the interpolation plus weighting method. The result shows that when the add-on lexicon was applied, the accuracy was increased about 2% compared to applying only SWN in positive-negative classification, while only 0.25% in 3-way classification. Therefore the add-on lexicon is more suitable for positive-negative sentiment classification than positive-neutral-negative sentiment classification. The reason may be that in the case of 3-way classification, some objective tweets were misclassified as subjective tweets when objective or OOV words were revised to subjective words.

Table 9 shows the performance of the add-on lexicon over the Sanders vs. SemEval 2013 corpus when using sentiment interpolation plus weighting method. It seems that the add-on lexicon performed better over the domain specific corpus (Sanders) than the general corpus (SemEval 2013). Using the add-on lexicon, the average accuracy of both 3-way and positive-negative classification tasks were improved by 1.49% on the Sanders corpus and 0.82% on the SemEval 2013 corpus.

Table 10 and Table 11 show examples of the revised positive and negative words with their POSs and scores obtained from the Sanders and SemEval 2013 corpora, respectively. It can be observed that the revised polarity words in the Sanders corpus are more domain-specific than those in the SemEval 2013 corpus since the Sanders corpus is a collection of tweets associated with only four keywords: Apple, Android, Microsoft and Twitter.

4.5 Comparison of Feature weigthing methods

Table 12 shows the comparison among four feature weighting methods and the baseline uni-gram. It reveals the average accuracy of the methods on both

Table 6: Results of 3-way classification task over the Sanders and SemEval 2013 corpora.

Methods		Sanders				SemEval 2013			
Feature	Lexicon	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
Uni-gram + POS	No	0.454	0.444	0.446	0.667	0.575	0.482	0.518	0.617
Sentiment Weighting	SWN	0.306	0.392	0.306	0.423	0.485	0.478	0.464	0.531
	+Addon	0.323	0.315	0.300	0.541	0.554	0.425	0.472	0.606
Sentiment Augmentation	SWN	0.496	0.452	0.471	0.690	0.611	0.487	0.536	0.628
	+Addon	0.485	0.452	0.466	0.684	0.620	**0.491**	0.542	0.635
Sentiment Interpolation	SWN	0.451	0.407	0.427	0.671	0.588	0.471	0.514	0.621
	+Addon	0.467	0.425	0.443	0.676	0.595	0.476	0.519	0.622
Sentiment Interpolation Plus	SWN	0.511	**0.439**	**0.471**	0.702	0.646	0.484	0.547	0.644
	+Addon	**0.522**	0.430	0.469	**0.705**	**0.650**	0.487	**0.550**	**0.646**

Table 7: Results of positive-negative classification task over the Sanders and SemEval 2013 corpora.

Methods		Sanders				SemEval 2013			
Feature	Lexicon	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
Uni-gram + POS	No	0.767	0.764	0.762	0.762	0.699	0.688	0.692	0.733
Sentiment Weighting	SWN	0.741	0.734	0.733	0.735	0.642	0.642	0.642	0.682
	+Addon	0.723	0.722	0.722	0.722	0.697	0.661	0.670	0.730
Sentiment Augmentation	SWN	0.776	0.773	0.771	0.771	0.719	0.700	0.707	0.750
	+Addon	0.765	0.763	0.762	0.762	0.725	0.712	0.717	0.755
Sentiment Interpolation	SWN	0.772	0.772	0.771	0.771	0.712	0.695	0.701	0.744
	+Addon	0.800	0.799	0.798	0.798	0.740	0.715	0.724	0.766
Sentiment Interpolation Plus	SWN	0.785	0.785	0.785	0.785	0.740	0.715	0.724	0.766
	+Addon	**0.813**	**0.812**	**0.812**	**0.812**	**0.759**	**0.728**	**0.739**	**0.780**

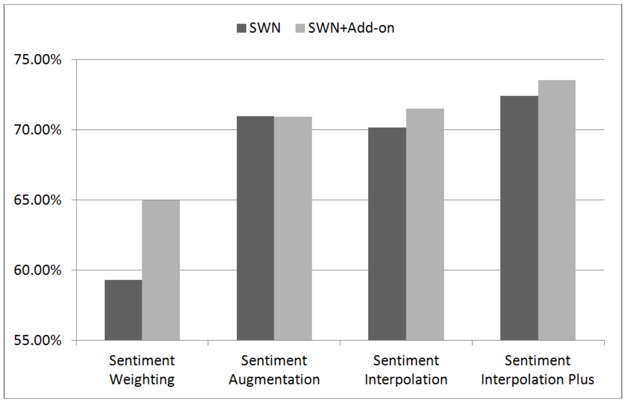


Figure 3: Average accuracy of SentiWordNet vs. SentiWordNet plus the add-on lexicon

Table 9: Performance of the add-on lexicon on the Sanders vs. SemEval 2013 corpus.

Corpus	SWN	+Add-on	Improvement
Sanders	74.34%	75.83%	1.49%
SemEval 2013	70.48%	71.30%	0.82%

Sanders and SemEval corpora in both 3-way classification and positive-negative classification tasks, where both SentiWordNet and the add-on lexicon are used as the sentiment lexicon. First, the accuracy of the sentiment weighting method (the score in the lexicon is used as the weight) was 4.51% worse than the uni-gram method. It may be because, unlike uni-gram weighting, the weights of objective and OOV words were set to 0 even when they appeared in the tweets. It means that the classifier loses the information about these words. Sentiment augmentation, where three lexicon scores were added to original uni-gram as extra features, improved the accuracy 1.43%. Sentiment interpolation, where lexicon scores were interpolated into uni-gram vector weights, further improved the accuracy 2.05% compared to baseline. Finally, the combination

Table 10: Examples of revised positive / negative words in the Sanders corpus.

Positive word	Revised score	Negative word	Revised score
#ics#OTHER	0.9223	battery#N	-0.9526
look#V	0.9211	customer#N	-0.9253
power#N	0.8926	update#N	-0.9109
:)#OTHER	0.8851	dear#OTHER	-0.9074
#android#N	0.8698	lot#N	-0.8931
help#V	0.8698	send#V	-0.8931
user#N	0.8664	#ios#OTHER	-0.8776
great#A	0.8252	service#N	-0.8049
game#N	0.8041	wait#V	-0.7434
thank#V	0.7994	ass#N	-0.7086

Table 11: Examples of revised positive / negative words in the SemEval 2013 corpus.

Positive word	Revised score	Negative word	Revised score
thank#V	0.8637	:(#OTHER	-0.9920
fun#A	0.8628	fuck#N	-0.9900
luck#N	0.8560	cancel#V	-0.9872
great#A	0.8442	damn#OTHER	-0.9864
:D#OTHER	0.8421	niggas#N	-0.9690
yay#OTHER	0.8341	die#V	-0.9554
pakistan#OTHER	0.8265	dont#V	-0.9329
:)#OTHER	0.8170	ass#N	-0.9272
yeah#OTHER	0.7999	cry#V	-0.9168
celebrate#V	0.7928	russia#OTHER	-0.9039

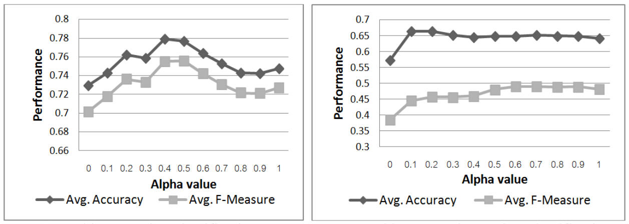
of sentiment interpolation and sentiment augmentation, called sentiment interpolation plus, achieved the highest accuracy among all methods with average accuracy improvement 4.08% compared to baseline uni-gram.

4.6 The sensitivity of α parameter

In the sentiment interpolation method, the α parameter in Equation (5) plays an important role for controlling the influence of uni-gram and sentiment lexicon scores. To analyze the effect of the α parameter, different values of the α parameter were applied. Note that when α is equal to 1, the vector weight becomes a fully uni-gram model (only term presence are used as feature weight) and when α is equal to 0, the vector weight value becomes a fully sentiment weighting model (only lexicon score are used as feature weight). Figures 4 a) and b) show the change of the average accuracy and F1-measure of the sentiment interpolation plus method on two datasets in the 3-way and positive-negative classification, respectively. In the positive-negative classification, the result clearly shows that the integration of uni-gram and lexicon score outperformed either uni-gram or sentiment weighting. The sentiment interpolation plus method performed well with large rage of α values (0.2 to 0.7). On the other hand, in the 3-way classification, it seems that the sentiment interpolation plus method only slightly increased the performance compared to uni-gram or sentiment weighting in most of the α values. As discussed earlier, the sentiment interpolation plus method was more suitable for the positive-negative classification than the 3-way classification task.

Table 12: Average accuracy comparison among four feature weighting methods and baseline uni-gram.

Methods	Avg. Acc	Improvement
Uni-gram + POS	69.49%	-
Sentiment Weighting	64.98%	-4.51%
Sentiment Augmentation	70.92%	1.43%
Sentiment Interpolation	**71.53%**	**2.05%**
Sentiment Interpolation Plus	**73.57%**	**4.08%**



(a) positive-negative classification

(a) positive-neutral-negative classification

Figure 4: Effect of the α parameter in the sentiment interpolation plus method

5 Conclusions

In this paper, we have shown an alternative hybrid method that incorporated sentiment lexicon information into the machine learning method to improve the performance of Twitter sentiment classification. There are two main contributions of this paper. First, we estimated the implicit polarity of objective and OOV words and used these words as additional information for the public sentiment lexicon. We described how we revised the polarity of objective and OOV words based on the assumption that the polarities of words are coincident with the polarity of their associated sentences, which seem reasonable due to the short length of tweets. Second, we proposed an alternative way to incorporate sentiment lexicon knowledge into a machine learning algorithm. We proposed the sentiment interpolation weighting method that interpolated lexicon score into uni-gram score in the feature vectors of SVM.

Our results indicate that the add-on lexicon improved the classification accuracy on average compared to using only the original public lexicon. The proposed sentiment interpolation weighting method performed well and the combination of sentiment interpolation and sentiment augmentation, called sentiment interpolation plus, with SentiWordNet and the add-on lexicon achieved the best performance and significantly improved the classification accuracy compared to the uni-gram model. The experiments show that the add-on lexicon performed better over the domain-specific corpus than the general corpus. In addition, our results indicate that the proposed approach was more appropriate for positive-negative classification than positive-neutral-negative (3-way) classification. Therefore, we plan to apply the subjective classification as our future work in order to filter the objective tweets before the polarity classification. Since negation words such as "not" and "less" are simply treated as uni-gram features in this work, another interesting issue is investigation on how special treatments of negation affect the polarity classification. Furthermore, we plan to find a method to reestimate the word polarity from unlabeled data or noisy label data instead of labeled data that is time consuming to create.

References

L. Barbosa and J. Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of Coling*.

F. Bravo-Marquez, M. Mendoza, and B. Poblete. 2013. Combining strengths, emotions and polarities for boosting Twitter sentiment analysis. In *Proceedings of WISDOM*.

A. Esuli, S. Baccianella, and F. Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of LREC*.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research, volume 9*

J. Fang and B. Chen. 2011. Incorporating Lexicon Knowledge into SVM Learning to Improve Sentiment Classification. In *Proceedings of IJCNLP*.

G. Y. Fausett. 1994. *Fundamentals of Neural Networks*. Prentice Hall PTR.

A. Go, R. Bhayani, and L. Huang 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.

C. Hung and H. Kai Lin. 2013. Using Objective Words in SentiWordNet to Improve Word-of-Mouth Sentiment Classification. *IEEE Intelligent Systems*, volume 28.

E. Kouloumpis, T. Wilson, and J. Moore. 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG!. In *Proceedings of ICWSM*.

A. Kumar and T. M. Sebastian 2012. Sentiment Analysis on Twitter *IJCSI*, volume 9.

B. Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing, volume 2*.

K. L. Liu , W. J. Li, and M. Guo 2012. Emoticon Smoothed Language Models for Twitter Sentiment Analysis. In *Proceedings of AAAI*.

A. Mudinas, D. Zhang, and M. Levene 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of WISDOM*.

P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of SemEval*.

T. O'Keefe and I. Koprinska 2009. Feature Selection and Weighting Methods in Sentiment Analysis. In *Proceedings of ADCS*.

A. Pak and P. Paroubek 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of LREC*.

H. Saif, M. Fernandez, Y. He, and H. Alani. 2012. Semantic sentiment analysis of twitter. In *Proceedings of ISWC*.

H. Saif, M. Fernandez, Y. He, and H. Alani. 2013. Evaluation Datasets for Twitter Sentiment Analysis. In *Proceedings of ESSEM*.

M. Thelwall, K. Buckley, and G. Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology, volume 9*.