

Incorporation of Target Specific Knowledge for Sentiment Analysis on Microblogging

Yongyos KAEWPITAKKUN[†], *Nonmember* and Kiyoaki SHIRAI^{†a)}, *Member*

SUMMARY Sentiment analysis of microblogging has become an important classification task because a large amount of user-generated content is published on the Internet. In Twitter, it is common that a user expresses several sentiments in one tweet. Therefore, it is important to classify the polarity not of the whole tweet but of a specific target about which people express their opinions. Moreover, the performance of the machine learning approach greatly depends on the domain of the training data and it is very time-consuming to manually annotate a large set of tweets for a specific domain. In this paper, we propose a method for sentiment classification at the target level by incorporating the on-target sentiment features and user-aware features into the classifier trained automatically from the data created for the specific target. An add-on lexicon, extended target list, and competitor list are also constructed as knowledge sources for the sentiment analysis. None of the processes in the proposed framework require manual annotation. The results of our experiment show that our method is effective and improves on the performance of sentiment classification compared to the baselines.

key words: *sentiment analysis, machine learning, microblogging, sentiment lexicon, polarity of target*

1. Introduction

Sentiment analysis and opinion mining are the fields of study that analyze people's opinions, sentiments, evaluations, attitudes, and emotions from written language [5]. Sentiment analysis of microblogging, e.g. Twitter, has become an important classification task because a great deal of user generated content is published on the Internet. Many researchers have adopted both machine learning and lexicon based approaches for sentiment analysis on Twitter. Most of these approaches aim at identifying the sentiment of the tweet, but not that addressed to a specific target in the tweet. In other words, they classify the sentiment at the tweet level, not at the target level. In Twitter, however, it is common that a user expresses several sentiments in one tweet or is expressing that sentiment about things other than the target and is neutral to the target. For example, the sentiment of the tweet "I hate when my mom annoying me with questions about her iphone" is clearly negative at the tweet-level, but neutral to the target "iPhone". These target-independent approaches may be insufficient for the practical use of sentiment analysis, since it is often required to know the sentiments towards a specific target, such as a product, brand,

or person. The users may want to know the opinion of other people about the products they are interested in, before making a purchase decision. Companies also want to know the overall opinion of their products of their potential users.

The goal of this research is to develop a method of classifying the sentiments (positive, negative or neutral) of a given target in the tweets. Our method relies on supervised machine learning. However, the proposed method does not require any human intervention, such as annotation of the labeled data. This enables us to apply our method to the sentiment analysis of various targets. Several techniques will be proposed to improve on the performance of target dependent sentiment classification. First, not general but target dependent training data is constructed for learning the sentiment classifier. It is automatically created by a lexicon-based method and several heuristics from unlabeled tweets. Second, a target-specific add-on lexicon is automatically constructed. A public sentiment lexicon is insufficient for target specific sentiment analysis, since the words used to express an opinion of the target are often not compiled in it. In this paper, an additional sentiment lexicon is built by automatically identifying the polarity of the objective and out-of-vocabulary words. Third, a user-aware feature is considered. The theory of Sentiment Consistency [6] indicates that the sentiment of two messages posted by the same user are more likely to be consistent than those of two randomly selected messages. Therefore, it would be better to take into consideration the other tweets of the same user that express an opinion about a given target. This user aware information can imply how likely are positive and negative opinions of the user to be expressed about the target. Finally, an extended target list and competitor list are introduced into the model. The former is the list of synonyms of the target. It is used to identify the target when expressed by different words or phrases. The latter is a list of the competitors of a given target (e.g. a product). People sometimes give their comments not only about the target itself but also its competitors, especially when they are comparing competing products. The competitor list can contribute to distinguishing whether the tweet is expressing an opinion about the target or its competitor.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 describes our proposed method and framework, including the data pre-processing, the creation of the add-on lexicon, competitor list creation, extended target creation, training data creation, and feature extraction. Section 4 describes the results of the experiments

Manuscript received July 1, 2015.

Manuscript revised November 6, 2015.

Manuscript publicized January 14, 2016.

[†]The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi-shi, 923-1292 Japan.

a) E-mail: kshirai@jaist.ac.jp

DOI: 10.1587/transinf.2015DAP0021

and presents a discussion. Finally, some conclusions and directions for future research are presented in Sect. 5.

2. Background and Related Work

There has been several attempts of sentiment analysis on general text such as product or movie review, focusing on the document level [1], sentence level [2], [3] and word and phrase level [4], [17]. Recently, the sentiment analysis on microblogging like Twitter is the upcoming trend in the current studies. However, classifying the sentiment of Twitter is different from the ordinary text and faces several challenges due to the short and informal language.

Early work on Twitter sentiment analysis used two approaches in traditional sentiment analysis: machine-learning based and lexicon-based approaches. The former employs supervised machine learning, such as Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machine (SVM) [7]–[9]. The problems with this approach are (1) it needs labeled training data, which requires much human labor, and (2) the classifier trained for one domain does not usually work well for another domain. On the other hand, the lexicon based approach uses pre-defined external resources, such as a polarity dictionary or a lexicon such as SentiWordNet[†], ANEW^{††} or MPQA^{†††} to determine the sentiment orientation in texts [10], [11]. The drawback of this approach is that it depends on pre-built lexicons and language models. Furthermore, most of the previous approaches aim at target-independent sentiment analysis, that is, classifying the sentiment at the tweet level, not the target level. However, as discussed in Sect. 1, a tweet can often have two or more sentiments towards multiple targets. Therefore, target independent approaches may be inappropriate because it is often required to classify the sentiment toward a certain topic. Chen examined whether a topic dependent model improves the polarity classification of microblogging [12]. They observed that, for some topics, topic dependent models achieved significantly better performance than a general model. Jiang incorporated target dependent features into the SVM classifier [13]. These features were extracted by rules based on syntactic relations in the result of the dependency parser. To boost the performance of the classification, they also used a graph based optimization by considering the sentiment labels of the related tweets. Dong proposed an Adaptive Recursive Neural Network (AdaRNN) for target dependent Twitter sentiment classification by propagating the sentiment of a word to the target based on the context and syntactic relationship in the dependency tree [14]. The common disadvantage of [13] and [14] is that it requires manually labeled training data for each specific topic. Moreover, the performance of their approach greatly depends on the dependency parser, which is not quite accurate when applied to informal language like

tweets. In our approach, the target-specific labeled training data is automatically created by a sophisticated method, not simply applying an existing sentiment classification tool to unlabeled data. Alternatively, some ways of performing unsupervised target-dependent sentiment analysis on Twitter have been proposed. Chen presented an optimization based approach to automatically extract sentiment expressions for a given target from a corpus of unlabeled tweets. Then, they applied a lexicon based method to classify the sentiment by summing up the score of the sentiment expression extracted from the previous step [15]. Zhang proposed an entity-level sentiment analysis for Twitter by combining lexicon based and learning based methods [16]. Their method first adopted a lexicon based approach to perform an entity-level sentiment analysis. After that, additional tweets that are likely to have opinions about the given entity were automatically identified through the Chi-square test based on the association between words and sentiment label of the tweets. Then, the classifier was trained to classify the polarity of the entities in the additional tweets extracted from the previous step.

In this paper, we present an alternative way to automatically create target-specific training data. We incorporate the procedures to (1) change the polarity of the tweets that are not truly related to the target into neutral, (2) invert the polarity of tweets expressing an opinion about a competitor of the target, by heuristic rules and (3) automatically construct target-specific add-on lexicon when we create the target-specific training data. This is used for training the SVM classifier with uni-gram, on-target sentiment and user-aware features for the prediction of the sentiment at the target level. Note that the use of user-aware features is one of the advantages of the proposed method. Another difference between our work and previous approaches ([15] and [16]) is how to identify the neutral tweets. Unlike previous work, where the neutral tweets are extracted by looking up indicator words obtained by statistical methods, in our method, they are identified by machine learning. Since the majority of the errors of target-dependent sentiment classification are caused by the tweets that show the user's opinion but not about the target [13], it would be appropriate to classify neutral-to-target tweets by sophisticated machine learning rather than a simple heuristic.

3. Proposed Method

This section presents the proposed approach. An overview of the system framework is shown in Fig. 1. For a given target, tweets containing the target word are retrieved by Twitter API. They are classified as positive, negative or neutral in several steps to create the target-specific training data. Then, two on-target features, sentiment and user-aware features, are extracted. Finally, the SVM is trained to classify the sentiment towards the target in the tweet.

3.1 Data Preprocessing

The data preprocessing process consists of part-of-speech

[†]<http://sentiwordnet.isti.cnr.it/>

^{††}http://neuro.imm.dtu.dk/wiki/A_new_ANEW/

^{†††}<http://mpqa.cs.pitt.edu/>

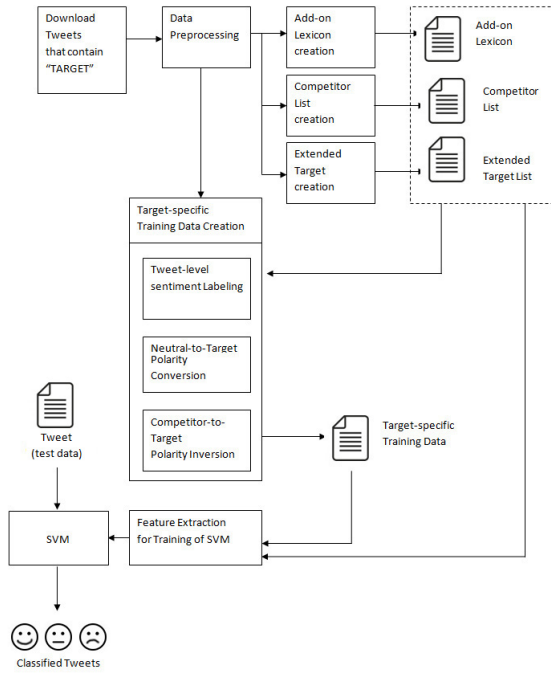


Fig. 1 System framework.

tagging, lemmatizing, and the removal of stop words and URLs. In the first step, the tweets are POS-tagged by the TweetNLP POS Tagger^{††††}, which is trained specially from Twitter data. Then, all words are lemmatized by the Stanford lemmatizer^{†††††}. We also reduce the number of letters that are repeated more than twice, e.g. “heellllooooo” is replaced by “hello”. Finally, the common stop words and URLs are removed because they represent neither a sentiment nor a semantic concept.

3.2 Creating a Target-Specific Add-On Lexicon

Due to its broad coverage, SentiWordNet [17] has become a famous and useful lexicon for sentiment analysis. However, more than 90 percent of the words in SentiWordNet are objective words, i.e., neither positive nor negative. Moreover, lots of words in tweets are slang, informal, or mistyped words that are not included in the lexicon [18]. In addition, the sentiment of the words may depend on the target domain. Based on this observation, we aim at constructing a target-specific add-on lexicon by compiling both objective and out-of-vocabulary (OOV) words with their newly estimated sentiment score from a corpus of unlabeled tweets.

After the pre-processing, the objective and OOV words of SentiWordNet are extracted from the retrieved tweets. In this step, only adjectives, interjections and hashtags are extracted, because they are the most informative for sentiment classification. For each word, the relevant tweets are retrieved by a query, the objective or OOV word and the given

target. The tweets that contain URLs are discarded because they commonly refer to some external resources, and re-tweet messages are also ignored because they are copies of the original tweets. Next, the polarity of each tweet is identified by the SentiStrength tool [19]. SentiStrength is a state-of-the-art lexicon based method for classifying the sentiment of short social texts, and it has been applied in much related research [20], [21]. Finally, the sentiment score of the objective and OOV words are calculated using Eq. (1). Note that $P(\text{positive}|w_i)$ and $P(\text{positive})$ are the ratio of the number of the tweets positively classified by SentiStrength to the number of the tweets containing w_i and all tweets, respectively. $P(\text{negative}|w_i)$ and $P(\text{negative})$ are defined in the same way.

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

where

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

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

Next, since scores in SentiWordNet are in the range of -1 to 1 , we have to revise our sentiment score in the same interval. In this study, a Bipolar sigmoid function [22] is used, as in Eq. (2).

$$Score(w_i)' = \text{sigmoid}(Score(w_i)) \quad (2)$$

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

The polarity score may be unreliable if the frequency of the word is too low, or the difference between the positive and the negative tendency is not large enough. Therefore, two thresholds are introduced. Threshold 1 (T_1) is the minimum number of words in the dataset and threshold 2 (T_2) 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 Eq. (3) is fulfilled. For the experiment in Sect. 4, we set the threshold T_1 to 10 and T_2 to 0.4, based on empirical observations.

$$\begin{aligned} & \text{Frequency of } w_i \text{ in dataset} \geq T_1 \\ & |Score_{POS}(w_i) - Score_{NEG}(w_i)| \geq T_2 \end{aligned} \quad (3)$$

3.3 Extended Target Creation

In Twitter, users might not express their opinion about a given target with the target keyword exactly. Sometimes, they comment about its features, concept, or things related

^{††††}<http://www.ark.cs.cmu.edu/TweetNLP/>

^{†††††}<http://nlp.stanford.edu/software/>

to the target. Therefore, it would be better to create an extended target list, consisting of terms that can be used as a representative of the target. For example, let us consider the tweet “I hate all Apple products”. It can be guessed that this user also hates the targets “iPhone”, “iPad” and “iPod”. So, the term “Apple” should be added to the extended target list of iPhone and so on. In [12], [13], the extended target list was created by measuring the Pointwise Mutual Information (PMI) between the candidate terms and the target on a corpus containing 20 million tweets. However, the performance of PMI is quite sensitive to the corpus size and it is very time-consuming to download a tweet corpus that contains enough data for the various candidate terms. Therefore, we propose a method to estimate the PMI using statistics obtained from Twitter API without a pre-downloaded tweet corpus.

First, the nouns and proper nouns are selected as the candidate terms of the extended targets. Next, we estimate the relatedness between the candidate term C and the target T by approximating the PMI as in Eq. (4). The functions $n(T)$, $n(C)$ and $n(T, C)$ are, respectively, the number of tweets containing T , C , and both T and C , while $time(T)$, $time(C)$ and $time(T, C)$ are the time ranges in which these tweets were posted. These statistics can be immediately obtained by TwitterAPI. $n(all_tweets)$ and $time(all_tweet)$ are estimated from the Twitter statistic that there are around 6,000 tweets per second on average[†]. The extended target list is built from all candidate terms whose PMI is greater than a threshold. For the experiment in Sect. 4, we set the threshold to 0 based on empirical observations.

$$PMI(T, C) = \log \frac{p(T, C)}{p(T)p(C)} = \log \frac{p(T|C)}{p(T)} \quad (4)$$

$$P(T|C) = \frac{\frac{n(T, C)}{time(T, C)}}{\frac{n(C)}{time(C)}}, \quad P(T) = \frac{\frac{n(T)}{time(T)}}{\frac{n(all_tweet)}{time(all_tweet)}}$$

3.4 Competitor List Creation

In Twitter, users might comment not only on the target itself, but also express their sentiment by comparing the target with its competitors. For example, a tweet “I’m fucking pissed I broke my iPhone and have to use this shitty Android” is clearly negative to “Android” and seems neutral to “iPhone”. If we know that Android is a competitor of iPhone, we can infer that this user is expressing their positive expression to the target “iPhone”. Therefore, it would be effective for sentiment analysis to create a list of the terms for which the sentiment is opposite to the target, called the competitor list. To the best of our knowledge, no general method to obtain the competitor-to-target terms has been reported. This paper presents a novel method to automatically create the list of terms that can be considered as competitors to the target by use of the word “VS” (versus) as the main keyword. It is usually used when people compare two things.

First, we build the queries “TARGET vs” and “vs TARGET” and enter them into the Search API of Twitter. Then, the retrieved set of tweets is cleaned by discarding the duplicate tweets and re-tweets, removing stop words and one character words. Next, we extract the two words connected immediately before or after the term “VS” as the candidate terms. More specifically, the terms are selected only when they are located on the opposite side of “VS” from the target term without “:”. Next, we measure the relatedness between the target and the candidate term by the PMI formula as shown in Eq. (4), and select as the competitors those terms where the PMI is greater than a threshold. For the experiment in Sect. 4, we set the threshold to 0 based on empirical observations. Finally, the terms in the extended target (described in Sect. 3.3) will be removed from the competitor list.

3.5 Target-Specific Training Data Creation

As discussed above, the performance of machine learning is sensitive to the domain of the training data. A classifier usually does not perform well when it is trained from the training data of a different domain [23]. This paper presents a novel method to create a target-specific training data set without manual annotation. We first use the state-of-the-art lexicon-based sentiment analysis tool that performs well at the tweet-level sentiment analysis. Then, we use heuristic rules to convert the sentiment to neutral if the sentiment score at the tweet level is very different from that at the target level. In other words, the sentiment of the tweets where the users express their opinion but not truly about the target will be converted to neutral. Finally, the sentiment labels of comparison tweets, where the users express their opinion of a competitor, will be inverted to the opposite orientation.

3.5.1 Tweet-Level Sentiment Labeling

In this step, we create a set of sentiment-classified tweets at the tweet level. Several researchers have used emoticons, such as :) and :(, or hashtags, such as #fail, to create data labeled with sentiments [24], [25]. However, both emoticons and hashtags are sparse for preparing a large amount of training data for some target keywords. In our proposed method, the tweets related to the target are first retrieved and pre-processed as described in Sect. 3.1. The tweets containing URLs or re-tweets are discarded from the data, since they could express a sentiment not about the target but about the contents of the linked page or other tweet. Then, similar to the add-on lexicon creation in Sect. 3.2, we use SentiStrength to classify the tweets as positive, negative or neutral.

3.5.2 Neutral-to-Target Polarity Conversion

The labels of the tweets in the corpus created by the previous step are the sentiments of the whole tweet, not the target.

[†]<http://www.internetlivestats.com/twitter-statistics/>

Algorithm 1: Neutral-to-target Polarity Conversion

Input: Tweet corpus with label TC_L , threshold T_1 , the SWN, the add-on lexicon, the extended target list

Output: Tweet corpus with new label of tweet TC_N

```

while not at end of the  $TC_L$  do
  read current tweet  $tw_i$ ;
  if  $tw_i$  contains more than one noun then
    for word  $w_i \in tw_i$  do
       $score_{tw} = score_{tw} + lexicon\_score(w_i)$ ;
      for target  $tg_i \in Extended\_Target\_List$  do
         $DWLS(w_i, tg_i) = \frac{lexicon\_score(w_i)}{distance(w_i, tg_i)}$ ;
      end
       $score_{tg} = score_{tg} + max_{tg_i}(DWLS(w_i, tg_i))$ ;
    end
    if  $\frac{|score_{tw} - score_{tg}|}{|score_{tw}|} > T_1$  then
      Set  $label(tw_i) = Neutral$ ;
    end
  end
end

```

We can usually regard the sentiment of the tweet as coinciding with that of the target, but the positive or negative tweets sometimes indicate a sentiment about other things, not the target. In such cases, the sentiment should be revised to neutral. Algorithm 1 shows the method of this neutral-to-target polarity conversion. First, two scores, called $score_{tw}$ and $score_{tg}$, are calculated by looking up in the public sentiment lexicon SentiWordNet and in our add-on lexicon. $score_{tw}$ is the summation of the lexicon score of all words in the tweet, called the “tweet-level lexicon score”. On the other hand, $score_{tg}$ is the summation of the lexicon score for the words that are probably related with the target or its extended target, called the “on-target lexicon score”. $DWLS(x, y)$ is the “distance weighted lexicon score” between the two words. It is defined as the lexicon score of x weighted by the reciprocal of the distance between x and y , where the distance is the length of the path from x to y in the dependency tree. Note that $score_{tg}$ is estimated based on $DWLS(w_i, tg_i)$. Finally, the sentiment label of the tweets will be converted to neutral if the relative difference between $score_{tw}$ and $score_{tg}$ is greater than a threshold. For the experiment in Sect. 4, we set the threshold to 0.66, based on empirical observations.

3.5.3 Competitor-to-Target Polarity Inversion

As discussed in Sect. 3.4, a user might express a sentiment by comparing the target with its competitors. If the sentiment label at the tweet level stands for the opinion about the competitors of the target, it should be inverted at the target level. Algorithm 2 shows the method of the competitor-to-target polarity inversion. First, we select only the tweets that contain the terms in the competitor list. Then, two scores, called $score_{tg}$ and $score_{cp}$, are calculated by looking up in SentiWordNet and our add-on lexicon. $score_{tg}$ is the “on-target lexicon score” in the neutral-to-target polarity conversion, and $score_{cp}$ is the summation of the lexicon score for the words that are probably related to the competitors, called

Algorithm 2: Competitor-to-target Polarity Inversion

Input: Tweet corpus with label TC_L , threshold T_2 , the SWN, the add-on lexicon, the extended target list, the competitor list

Output: Tweet corpus with new label of tweet TC_N

initialization;

```

while not at end of the  $TC_L$  do
  read current tweet  $tw_i$ ;
  if  $tw_i$  contains competitors then
    for word  $w_i \in tw_i$  do
      for target  $tg_i \in Extended\_Target\_List$  do
         $DWLS(w_i, tg_i) = \frac{lexicon\_score(w_i)}{distance(w_i, tg_i)}$ ;
      end
       $score_{tg} = score_{tg} + max_{tg_i}(DWLS(w_i, tg_i))$ ;
      for competitor  $cp_i \in Competitor\_List$  do
         $DWLS(w_i, cp_i) = \frac{lexicon\_score(w_i)}{distance(w_i, cp_i)}$ ;
      end
       $score_{cp} = score_{cp} + max_{cp_i}(DWLS(w_i, cp_i))$ ;
    end
    if  $|score_{tg} - score_{cp}| > T_2$  and  $sign(tw_i) = sign(score_{cp})$  and  $sign(score_{tg}) \neq sign(score_{cp})$  then
      Set  $label(tw_i) = opposite(label(tw_i))$ ;
    end
  end
end

```

the “on-competitor lexicon score”. Both $score_{tg}$ and $score_{cp}$ are calculated by the lexical score weighted by the reciprocal of the distance in the dependency tree. The sentiment label will be inverted if the difference between $score_{tg}$ and $score_{cp}$ is greater than a threshold. Moreover, the polarity of the tweet should be inverted only when the main opinion is expressed about the competitor. More specifically, the tweet label will be inverted only when the sign of the original label (denoted by $sign(tw_i)$ in Algorithm 2, determined in the process in Sect. 3.5.1) is the same as the sign of $score_{cp}$, and the sign of $score_{tg}$ and $score_{cp}$ is not the same. For the experiment in Sect. 4, we set the threshold to 0.2, based on empirical observations.

3.6 Feature Extraction

In this subsection, we will explain how to represent a tweet as a feature vector to train a classifier for target-dependent sentiment classification.

3.6.1 Uni-gram and POS Features

Uni-gram and POS features are common and widely used in the domain of sentiment analysis. Although there are many feature weighting schemes for uni-gram, binary weighting is used as the baseline method in this work. That is, the weights of a word (POS) is 1 if it is present in the tweet, otherwise 0.

3.6.2 On-Target Sentiment Features

The polarity score of the sentiment lexicon is widely used

as a feature in sentiment classification, too. In order to perform a sentiment analysis at the target level, *on-target_lexicon_score* of the sentiment words in both SentiWordNet and the add-on lexicon are defined by Eq. (5). tg' is the closest target (or its extended target) to the sentiment word w_i , while cp' is the closest term in the competitor list. The score is weighted by the reciprocal of the distance between the sentiment word w_i and tg' if tg' is closer than cp' , otherwise between w_i and cp' with sign inversion. The sentiment words in the tweet are then classified into two classes: positive and negative, based on their score. On-target sentiment features are two additional features for positive and negative classes whose weights are defined as the sum of the *on-target_lexicon_score* of the positively and negatively classified words.

On-target_lexicon_score($w_i, target$)

$$= \begin{cases} DWLS(w_i, tg'), & \text{if } DWLS(w_i, tg') \geq DWLS(w_i, cp') \\ (-1) \times DWLS(w_i, cp'), & \text{otherwise.} \end{cases} \quad (5)$$

3.6.3 User-Aware Features

User-aware features represent the latent opinion of the user about a given target. It is often difficult to understand the opinion of the user from one short tweet. In our method, other tweets of the user are taken into account to guess the user's latent opinion of the target. SentiStrength is used to classify the tweets which contain the target word and are posted by the same user, as either positive, negative, or neutral. The user-aware features are three additional features for positive, negative and neutral classes whose weights are defined as the percentage of the positive, negative and neutral tweets of the user.

4. Evaluation

4.1 Dataset

Because people usually express their opinion about products, brands, companies and celebrities, we selected "iPhone", "Xbox", "Nike" (products/brands), "Google", "Verizon", "Sony" (companies) and "Obama", "Beyonce", "Messi" (person) as the targets for sentiment analysis. In order to create the training data, we downloaded, via the Twitter Search API, the collection of those tweets that contain the target keyword. After the creation of the target-specific training dataset (as in Sect. 3.5), we balanced the number of positive, negative and neutral tweets so that the training data would consist of equal numbers of tweets for each class. Due to the limitation of Twitter API[†], we select one representative target for each domain to evaluate the effectiveness of the user-aware feature. For each user in the training data,

[†]The limit for getting timeline tweets of the users is set to 180 requests per 15 minutes.

Table 1 Statistics of the dataset.

| Target | Tweets in training set | Tweets in test set | Users | On-target tweets |
|---------|------------------------|--------------------|--------|------------------|
| iPhone | 10,500 | 300 | 10,500 | 64,260 |
| Xbox | 15,000 | 300 | - | - |
| Nike | 15,000 | 300 | - | - |
| Google | 12,000 | 300 | 12,000 | 69,240 |
| Verizon | 15,000 | 300 | - | - |
| Sony | 9,000 | 300 | - | - |
| Obama | 10,500 | 300 | 10,500 | 258,510 |
| Beyonce | 13,500 | 300 | - | - |
| Messi | 13,500 | 300 | - | - |
| Total | 114,000 | 2,700 | 33,000 | 392,010 |

3,200 tweets posted by that user were downloaded. Then, only the tweets containing the target (on-target tweets) were used to obtain the user-aware feature. For the test data, another 300 tweets (100 for each class) including the target keyword were retrieved. They were manually annotated with the sentiment for the target. Statistics of the dataset is shown in Table 1.

4.2 Results and Discussion

We conducted several experiments to evaluate the effectiveness of our proposed method. The average of F1 measure (harmonic mean of the precision and recall) over the sentiment classes as well as accuracy are used as our evaluation criteria. The performance of the following methods was measured.

Sentiment140^{††}: a supervised method that discovers the current sentiment for a brand, product, or topic on Twitter, developed by graduate students at Stanford University. This is the baseline.

SentiStrength: a state-of-the-art lexicon based method for classifying the sentiment of short social texts. This is another baseline.

SVM-SS: an SVM classifier trained from the training data labeled by SentiStrength with uni-gram features. No other technique described in Sect. 3 was applied.

SVM-Our: an SVM classifier trained from our target-specific training data described in Sect. 3.5 with uni-gram features.

SVM-Our_Sen: an SVM classifier trained from our target-specific training data with uni-gram and on-target sentiment features.

SVM-Our_Usr: an SVM classifier trained from our target-specific training data with uni-gram and user-aware features.

SVM-Our_All: an SVM classifier trained from our target-specific training data with all the features described in this paper.

We used LIBLINEAR [26] (L2-regularized L2-loss support vector classification) for training the SVM classifiers. The regularization parameter c was optimized by cross-validation on the training data.

^{††}<http://help.sentiment140.com/api/>

Table 2 Results of one-step classification.

| | Sentiment140 | | SentiStrength | | SVM-SS | | SVM-Our | | SVM-Our_Sen | | SVM-Our_Usr | | SVM-Our_All | |
|-------------|--------------|-------|---------------|-------|--------|-------|---------|-------|--------------|--------------|-------------|--------------|--------------|--------------|
| Target | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC |
| iPhone | 0.449 | 0.457 | 0.602 | 0.540 | 0.598 | 0.560 | 0.622 | 0.587 | 0.635 | 0.587 | 0.631 | 0.593 | 0.633 | 0.587 |
| Xbox | 0.570 | 0.523 | 0.718 | 0.603 | 0.724 | 0.637 | 0.709 | 0.633 | 0.729 | 0.663 | - | - | - | - |
| Nike | 0.489 | 0.433 | 0.682 | 0.593 | 0.696 | 0.620 | 0.684 | 0.610 | 0.704 | 0.650 | - | - | - | - |
| Google | 0.488 | 0.483 | 0.606 | 0.517 | 0.648 | 0.583 | 0.646 | 0.590 | 0.665 | 0.607 | 0.647 | 0.587 | 0.676 | 0.617 |
| Verizon | 0.562 | 0.510 | 0.679 | 0.600 | 0.708 | 0.627 | 0.719 | 0.660 | 0.723 | 0.660 | - | - | - | - |
| Sony | 0.551 | 0.530 | 0.622 | 0.557 | 0.676 | 0.607 | 0.659 | 0.603 | 0.682 | 0.623 | - | - | - | - |
| Obama | 0.301 | 0.383 | 0.526 | 0.447 | 0.550 | 0.493 | 0.526 | 0.493 | 0.559 | 0.503 | 0.544 | 0.500 | 0.556 | 0.510 |
| Beyonce | 0.412 | 0.390 | 0.670 | 0.563 | 0.686 | 0.583 | 0.696 | 0.600 | 0.703 | 0.607 | - | - | - | - |
| Messi | 0.469 | 0.453 | 0.667 | 0.567 | 0.660 | 0.573 | 0.672 | 0.603 | 0.682 | 0.607 | - | - | - | - |
| Average (3) | 0.412 | 0.441 | 0.578 | 0.501 | 0.599 | 0.546 | 0.598 | 0.557 | 0.620 | 0.566 | 0.607 | 0.560 | 0.622 | 0.571 |
| Average (9) | 0.477 | 0.463 | 0.641 | 0.554 | 0.661 | 0.587 | 0.659 | 0.598 | 0.676 | 0.612 | - | - | - | - |

Table 3 Results of two-step classification.

| | Sentiment140 | | SentiStrength | | SVM-SS | | SVM-Our | | SVM-Our_Sen | | SVM-Our_Usr | | SVM-Our_All | |
|-------------|--------------|-------|---------------|-------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------|--------------|--------------|
| Target | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC | F1 | ACC |
| iPhone | 0.449 | 0.457 | 0.602 | 0.540 | 0.603 | 0.553 | 0.619 | 0.563 | 0.638 | 0.577 | 0.623 | 0.560 | 0.636 | 0.573 |
| Xbox | 0.570 | 0.523 | 0.718 | 0.603 | 0.721 | 0.633 | 0.718 | 0.643 | 0.721 | 0.650 | - | - | - | - |
| Nike | 0.489 | 0.433 | 0.682 | 0.593 | 0.696 | 0.617 | 0.675 | 0.590 | 0.683 | 0.610 | - | - | - | - |
| Google | 0.488 | 0.483 | 0.606 | 0.517 | 0.631 | 0.560 | 0.641 | 0.570 | 0.674 | 0.603 | 0.633 | 0.557 | 0.659 | 0.580 |
| Verizon | 0.562 | 0.510 | 0.679 | 0.600 | 0.717 | 0.647 | 0.704 | 0.643 | 0.704 | 0.643 | - | - | - | - |
| Sony | 0.551 | 0.530 | 0.622 | 0.557 | 0.676 | 0.590 | 0.651 | 0.580 | 0.670 | 0.600 | - | - | - | - |
| Obama | 0.301 | 0.383 | 0.526 | 0.447 | 0.545 | 0.483 | 0.532 | 0.480 | 0.532 | 0.470 | 0.548 | 0.477 | 0.552 | 0.487 |
| Beyonce | 0.412 | 0.390 | 0.670 | 0.563 | 0.683 | 0.577 | 0.689 | 0.583 | 0.702 | 0.597 | - | - | - | - |
| Messi | 0.469 | 0.453 | 0.667 | 0.567 | 0.649 | 0.560 | 0.661 | 0.590 | 0.602 | 0.583 | - | - | - | - |
| Average (3) | 0.412 | 0.441 | 0.578 | 0.501 | 0.593 | 0.532 | 0.597 | 0.538 | 0.615 | 0.550 | 0.601 | 0.531 | 0.616 | 0.547 |
| Average (9) | 0.477 | 0.463 | 0.641 | 0.554 | 0.658 | 0.580 | 0.655 | 0.583 | 0.665 | 0.593 | - | - | - | - |

4.2.1 Results of the Sentiment Analysis at the Target Level

In this experiment, the tweets were classified into positive, neutral, or negative about the target. Two approaches were evaluated. First, the tweets were classified as positive, neutral or negative in a single step. Second, a two-step classification was performed: the tweets were classified as subjective or neutral to the target in step 1, then the subjective tweets were classified to positive or negative in step 2. Tables 2 and 3 show the results of the one-step and two-step classifications. The row ‘Average(3)’ and ‘Average (9)’ indicate the average of three targets where the user-aware feature is used and all nine targets, respectively.

The results show that, on average, the one-step classification was slightly better than the two-step classification. This was because of the low recall (0.27–0.44) and F1 measure (0.36–0.49) of the neutral class in the first step of the two-step classification. This meant that the classification of the subjectivity was rather difficult. With the one-step classification, we found that our methods (SVM-Our_Sen) outperformed the two baselines (Sentiment140 and SentiStrength) by large margins and improved the accuracy over the two baselines by 5.7%–14.9%, respectively. Our methods also improved the performance compared to SVM-SS (where the training data was labeled by SentiStrength only) about 2.5%. Table 4 shows P-values of McNemar’s test to evaluate the significance of the differences between SVM-Our_Sen and three baselines in the one-step classification. Comparing the results of individual targets, our method significantly

Table 4 Statistical test of the difference between SVM-Our_Sen and the baseline.

| Target | The two-tailed, P-Value | | |
|---------|-------------------------|---------------|---------------|
| | Sentiment140 | SentiStrength | SVM-SS |
| iPhone | 0.0007 | 0.0660 | 0.2299 |
| Xbox | 0.0002 | 0.0125 | 0.2963 |
| Nike | 0.0001 | 0.0472 | 0.2531 |
| Google | 0.0014 | 0.0009 | 0.3239 |
| Verizon | 0.0001 | 0.0207 | 0.5708 |
| Sony | 0.0197 | 0.0251 | 0.1003 |
| Obama | 0.0038 | 0.0611 | 0.7656 |
| Beyonce | 0.0001 | 0.1048 | 0.2482 |
| Messi | 0.0001 | 0.1344 | 0.1649 |
| All | 0.0001 | 0.0001 | 0.0003 |

outperformed Sentiment140 and SentiStrength at 99% and 90% confident level for the most targets, respectively. SVM-Our_Sen was better than SVM-SS for all 9 targets by 0.5–3.7% F1 measure and 0.1–3.4% accuracy, although the differences were not so significant. Comparing all targets, our method outperformed each baseline at 99% confident level.

The user-aware feature (SVM-Our_Usr) performed well and the combination of both features (SVM-Our_All) achieved the highest performance in average of 3 representative targets. Moreover, the accuracy for Obama was lower than other targets, because many tweets about Obama contain a lot of sarcasm and irony, which requires special techniques. In addition, the detailed analysis of the on-target sentiment and user-aware features will be shown in 4.2.4.

4.2.2 Evaluation of the Subjective and Polarity Classification

We conducted two experiments to evaluate the effectiveness of our proposed method for subjectivity and polarity classification tasks. In the subjectivity classification, we considered positive and negative tweets as a subjective class. Note that the balanced tweet corpus consisting of equal numbers of subjective and neutral tweets was used as the training and test data, unlike step 1 of the two-step classification in Sect. 4.2.1. On the other hand, in the polarity classification, the neutral tweets were discarded, and the tweets were classified as positive or negative. Tables 5 and 6 show the F1 measure of the subjectivity and polarity classification. The results clearly show that our proposed method (SVM-Our.Sen) outperformed the uni-gram model (SVM-SS) by 3.5% in the subjectivity classification task and 1.4% in the polarity classification task when considering all 9 targets. These results indicate that our method is more effective for the subjectivity classification task than the polarity classification task, as we had expected. Our method was mainly designed to distinguish between the tweets that expressed an opinion but not truly about the target, which should be classified as neutral at the target level. In addition, the on-target sentiment feature performed better than the user-aware feature, and the combination of both features achieved the highest performance in both the subjectivity and polarity classification tasks in the average of 3 representative targets.

4.2.3 Contribution of the Add-On Lexicon, Extended Target List, and Competitor List

We evaluated the contribution of three target specific data sets: the add-on lexicon, the extended target list, and the competitor list. Table 7 compares the methods without one of these three extra data sets, the method with all of them (SVM-Our.Sen), and the baseline trained with only the uni-gram feature (SVM-SS). It shows the average accuracy of three and all categories. The competitor list seems the most useful for a product like the iPhone, while the extended target list performs the best for a company, such as Google. This may be caused by the fact that people usually compare a product with its competitors, while a company has a lot of features or extended targets compared to a product. In addition, the add-on lexicon, where the polarities of the objective and OOV words were estimated, made the highest contribution on average. As discussed in Sect. 3.2, since there are a lot of objective and OOV words in informal text such as tweets, the add-on lexicon can supply the necessary information for a target-level sentiment analysis. Examples of an add-on lexicon, extended target list, and competitor list are shown in Tables 8, 9 and 10, respectively. One can see that many hashtags that can be used with the given target were added to the add-on lexicon. Most of the extended targets and competitors also seem reasonable.

Table 5 F1 measure of subjectivity classification.

| Target | SVM-SS | SVM-Our.Sen | SVM-Our.Usr | SVM-Our.All |
|-------------|--------|--------------|--------------|--------------|
| iPhone | 0.608 | 0.664 | 0.680 | 0.670 |
| Xbox | 0.570 | 0.616 | - | - |
| Nike | 0.616 | 0.623 | - | - |
| Google | 0.577 | 0.627 | 0.607 | 0.613 |
| Verizon | 0.719 | 0.732 | - | - |
| Sony | 0.561 | 0.617 | - | - |
| Obama | 0.539 | 0.558 | 0.538 | 0.577 |
| Beyonce | 0.589 | 0.606 | - | - |
| Messi | 0.659 | 0.709 | - | - |
| Average (3) | 0.575 | 0.617 | 0.608 | 0.620 |
| Average (9) | 0.604 | 0.639 | - | - |

Table 6 F1 measure of polarity classification.

| Target | SVM-SS | SVM-Our.Sen | SVM-Our.Usr | SVM-Our.All |
|-------------|--------------|--------------|-------------|--------------|
| iPhone | 0.776 | 0.807 | 0.792 | 0.802 |
| Xbox | 0.885 | 0.885 | - | - |
| Nike | 0.84 | 0.845 | - | - |
| Google | 0.775 | 0.817 | 0.787 | 0.808 |
| Verizon | 0.835 | 0.82 | - | - |
| Sony | 0.815 | 0.825 | - | - |
| Obama | 0.698 | 0.691 | 0.711 | 0.712 |
| Beyonce | 0.815 | 0.846 | - | - |
| Messi | 0.794 | 0.823 | - | - |
| Average (3) | 0.750 | 0.772 | 0.763 | 0.774 |
| Average (9) | 0.804 | 0.818 | - | - |

Table 7 Contribution of 3 target-specific data.

| Method | AVG (Product) | AVG (Company) | AVG (Person) | AVG (ALL) |
|---------------------|---------------|---------------|--------------|--------------|
| SVM-SS | 0.673 | 0.677 | 0.632 | 0.661 |
| + ALL (SVM-Our.Sen) | 0.689 | 0.690 | 0.648 | 0.676 |
| - Extended Target | 0.686 | 0.677 | 0.643 | 0.668 |
| - Competitor List | 0.678 | 0.687 | 0.645 | 0.670 |
| - Add-on Lexicon | 0.680 | 0.686 | 0.636 | 0.667 |

4.2.4 Contribution of On-Target Sentiment and User-Aware Features

Table 11 shows the average F1 measure when the on-target sentiment features and (non-target-specific) sentiment feature were used. The on-target sentiment feature is derived from the weighted sum of the scores of the sentiment words as in Eq. (5), where the weights are defined as the distance between the sentiment words and the target or competitor, while the sentiment feature is derived from the non-weighted score of the sentiment lexicon. The results reveal that the on-target sentiment feature helps the classifier to improve the performance for the target-level sentiment classification in all tasks. Improvements of 2.1% and 1.1% are found in the subjectivity and polarity classification, respectively, which are consistent with the results in Tables 5 and 6. This is because some polarity words that do not truly express a sentiment about the target are less considered in the model with the on-target sentiment feature.

Table 8 Examples of words in the add-on lexicon.

| iPhone | | Google | | Obama | |
|----------------|------------------|---------------|-----------------|----------|---------------|
| Positive | Negative | Positive | Negative | Positive | Negative |
| #beautiful | #unhappycustomer | wowza | #nervous | #likes | #wakepamerica |
| #lovers | freezing | #awesomesauce | #fuck | bless | #rapist |
| amo | #fuck | okaay | google-worst | #smiles | #illegal |
| #greatmusic | #problems | #googleedu | #translate | hahahaa | blind |
| #teamapple | #iphoneproblems | #greatproduct | #annoyingthings | #heroic | #dictator |
| #app | #frustrated | #search | #ridiculous | #saved | #terrorism |
| #wickedawesome | #autocorrect | hihihi | #torture | #fashi | #radicalislam |

Table 9 Examples of the words in the extended target list.

| iPhone | Google | Obama |
|----------|------------|----------------|
| chargers | fiber | america |
| ipod | app | action |
| battery | play | michelle |
| ios | translator | speech |
| apple | search | policy |
| ipad | android | administration |
| itunes | news | americans |
| app | store | threat |
| sprint | chrome | pres |
| cable | maps | president |
| charger | nexus | barack |

Table 10 Examples of the words in the competitor list.

| iPhone | Google | Obama |
|------------|------------|-------------|
| droid | mozilla | bush |
| android | apple | mitt |
| samsung | xiaomi | bibi |
| galaxy | alibaba | congress |
| htc | duckduckgo | putin |
| blackberry | firefox | romney |
| xperia | searchblox | walker |
| sony | bing | gop |
| nexus | venmo | netanyahu |
| google | penguin | guiliani |
| moto | cyanogen | republicans |

Table 11 Evaluation of the on-target sentiment features.

| Task | On-target sentiment features | Sentiment features |
|---------------------------|------------------------------|--------------------|
| 3-class classification | 0.676 | 0.661 |
| Subjective classification | 0.639 | 0.618 |
| Polarity classification | 0.818 | 0.807 |

Table 12 Evaluation of the user-aware features.

| | iPhone | Google | Obama |
|-------------------------------------|-------------|--------|-------------|
| Avg no. of on-target tweet per user | 6.12 | 5.77 | 24.62 |
| Avg F-1 improvement | 0.8% | 0.1% | 1.8% |

Table 12 shows the average number of on-target tweets per user and the average difference in F1 measure between the model with the user-aware feature (SVM-Our_Usr) and without (SVM-Our). The user-aware feature was able to improve the F1 measure for a product (iPhone) and person (Obama) but not for the company (Google). We guess there are two major reasons. First, the sentiments of people might be more consistent for product or people entities than for a company. People who have a positive or nega-

tive feeling about some product or person usually express the same sentiment about it in their tweet collection. On the other hand, a person might express difference sentiments about a company due to the variety of aspects of that company. For example, the user may express a positive sentiment about Google's search engine but a negative sentiment about Google's translator. Therefore, the sentiment of the user may not be consistent for a company, especially a big company like Google. Second, the performance of the user-aware feature also depends on the number of on-target tweets of each user. Because of the limitations of Twitter API, we can download only the last 3,200 tweets of each user. Note that the number of tweets containing the target keyword is much smaller than the limitation, as shown in Table 1. Intuitively, the user-aware feature is less reliable when the size of the tweet corpus is small. Actually, the improvement about Obama, where there are 24.62 tweets per user on average, is greater than for iPhone, where there are only 6.12 tweets. Therefore, other information, i.e. the friendship networks of the users, should be considered to overcome the sparseness of the data of the users' tweets and improve the performance of the user-aware feature.

5. Conclusions

In this paper, we have presented an alternative method for incorporating on-target sentiment information and user-aware information into a machine learning classifier for the sentiment analysis of the tweets at the target level. Our method requires no human annotation for the development of the classifier. First, three extra resources, an add-on lexicon, an extended target list, and a competitors list, were automatically constructed from the unlabeled tweets. Then, target-specific training data was created based on heuristic rules and the lexicon-based sentiment analysis method. Two new features for training the sentiment classifier were introduced. One is the on-target sentiment feature, giving greater weight to the sentiments of the words near the target; the other is the user-aware feature, that captures the tendency of the sentiment expressed by the same user. The results of the experiment indicate that our proposed method is effective and improves the classification accuracy compared to the baseline methods in both the 3-class classification and the subjectivity/polarity classification.

The contribution of the user-aware feature is not so marked, because it is difficult to prepare a large amount of the tweets posted by a user. In the future, we plan to incor-

porate other network information, i.e. the social relations of the users, to overcome this problem and improve the performance of the user-aware feature. Furthermore, we plan to find a sophisticated method to retrieve the relevant tweets and filter the spam and advertising tweets before the polarity classification instead of simply using the URLs as an indicator.

References

- [1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol.2, no.1–2, pp.1–135, 2008.
- [2] H. Yu, and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," *Proc. 2003 conference on Empirical methods in natural language processing*, pp.129–136, Association for Computational Linguistics, 2003.
- [3] S.-M. Kim, and E. Hovy, "Determining the sentiment of opinions," *Proc. 20th international conference on Computational Linguistics*, Article No.1367, Association for Computational Linguistics, 2004.
- [4] V. Hatzivassiloglou, and K. McKeown, "Predicting the semantic orientation of adjectives," *Proc. 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pp.174–181, Association for Computational Linguistics, 1997.
- [5] B. Liu, "Sentiment analysis and subjectivity," *Handbook of natural language processing*, vol.2, pp.627–666, 2010.
- [6] R.P. Abelson, "Whatever became of consistency theory?," *Personality and Social Psychology Bulletin*, vol.9, no.1, pp.37–54, 1983.
- [7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *Proc. ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp.79–86, Association for Computational Linguistics, 2002.
- [8] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," *Proc. 23rd International Conference on Computational Linguistics: Posters*, pp.36–44, Association for Computational Linguistics, 2010.
- [9] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," *LREC*, pp.1320–1326, 2010.
- [10] X. Ding, B. Liu, and P.S. Yu, "A holistic lexicon-based approach to opinion mining," *Proc. 2008 International Conference on Web Search and Data Mining*, pp.231–240, ACM, 2008.
- [11] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol.37, no.2, pp.267–307, 2011.
- [12] F. Chen and S.H. Mirisae, "Do topic-dependent models improve microblog sentiment estimation?," *Proc. ICWSM, AAAI*, pp.575–578, 2014.
- [13] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp.151–160, Association for Computational Linguistics, 2011.
- [14] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," *Proc. 52nd Annual Meeting of the Association for Computational Linguistics*, pp.49–54, 2014.
- [15] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A.P. Sheth, "Extracting diverse sentiment expressions with target-dependent polarity from twitter," *ICWSM*, pp.50–57, 2012.
- [16] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexiconbased and learning-based methods for twitter sentiment analysis," *HP Laboratories, Technical Report HPL-2011*, vol.89, 2011.
- [17] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," *LREC*, pp.2200–2204, 2010.
- [18] Y. Kaewpitakkun, K. Shirai, and M. Mohd, "Sentiment lexicon interpolation and polarity estimation of objective and out-of-vocabulary words to improve sentiment classification on microblogging," 2014.
- [19] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology*, vol.63, no.1, pp.163–173, 2012. <http://sentistrength.wlv.ac.uk/>
- [20] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," *Proc. First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, Article No.5, pp.1–8, ACM, 2012.
- [21] F. Ren and Y. Wu, "Predicting user-topic opinions in twitter with social and topical context," *Affective Computing, IEEE Transactions on*, vol.4, no.4, pp.412–424, 2013.
- [22] L.V. Fausett and P. Hall, *Fundamentals of neural networks: architectures, algorithms, and applications*, Prentice-Hall Englewood Cliffs, 1994.
- [23] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," *Proc. Recent Advances in Natural Language Processing (RANLP)*, pp.2–1, Citeseer, 2005.
- [24] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report*, Stanford, pp.1–12, 2009.
- [25] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *ICWSM*, vol.11, pp.538–541, 2011.
- [26] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," *J. Machine Learning Research*, vol.9, pp.1871–1874, 2008. <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>



Yongyos Kaewpitakkun received M.S degree from Asian Institute of Technology in 2002. He was a research assistant at Asian Institute of Technology of Technology from 2003 to 2005. He is now a PhD student of Japan Advanced Institute of Science and Technology under the Doctoral Research Fellow. His research interests include data mining, sentiment analysis and personalization.



Kiyoaki Shirai received the B.S (1993), M.S (1995) and D.S (1998) degrees in Engineering from Tokyo Institute of Technology. He was a research assistant at Tokyo Institute of Technology from 1998 to 2001. He has been an associated professor at School of Information Science, Japan Advanced Institute of Science and Technology since 2001. His research interests lie in the fields of natural language processing.