Target-Dependent Twitter Sentiment Classification with Rich Automatic Features

Duy-Tin Vo and **Yue Zhang**
Singapore University of Technology and Design
8 Somapah Road, Singapore 487372
duytin_vo@mymail.sutd.edu.sg, yue_zhang@sutd.edu.sg

Abstract

Target-dependent sentiment analysis on Twitter has attracted increasing research attention. Most previous work relies on syntax, such as automatic parse trees, which are subject to noise for informal text such as tweets. In this paper, we show that competitive results can be achieved without the use of syntax, by extracting a rich set of automatic features. In particular, we split a tweet into a left context and a right context according to a given target, using distributed word representations and neural pooling functions to extract features. Both sentiment-driven and standard embeddings are used, and a rich set of neural pooling functions are explored. Sentiment lexicons are used as an additional source of information for feature extraction. In standard evaluation, the conceptually simple method gives a 4.8% absolute improvement over the state-of-the-art on three-way targeted sentiment classification, achieving the best reported results for this task.

1 Introduction

As a popular channel for sharing opinions and feelings, Tweets have become an important domain for sentiment analysis (SA) research over the past few years. While seminal work studied the sentiment of whole Tweets [Go *et al.*, 2009; Davidov *et al.*, 2010; Pak and Paroubek, 2010; Mohammad *et al.*, 2013], **target-dependent SA on tweets** has gained increasing attention [Jiang *et al.*, 2011; Mitchell *et al.*, 2013; Dong *et al.*, 2014]. The task is to categorize the sentiment towards particular targets in a tweet (i.e. positive, negative or neutral), predicting exactly which object bares what specific opinion. For example, in "Windows is much better than OS X!", "Windows" is reflected in positive sentiment, while "OS X" receives the opposite sentiment.

Jiang et al. [2011] was the first to propose targeted SA on Twitter, who emphasize the importance of targets by showing that 40% of SA errors are caused by not considering them in classification. They incorporate 7 rule-based target-dependent features into a model with traditional target-independent SA features, which give a significant improvement. Further along the line, Mitchell et al. [2013] apply a sequence labeling model to simultaneously detect entities and predict opinions towards them. Recently, Dong et al. [2014] propose an adaptive recursive neural network for target-dependent Twitter sentiment classification, which propagates sentiment signals from sentiment-baring words to specific targets on a dependence tree.

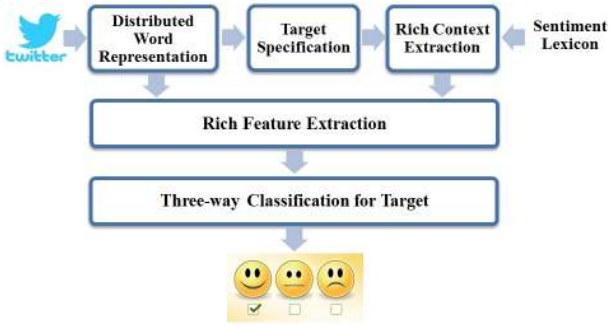

Figure 1: System Architecture.

All the methods above rely on syntax (e.g. parse trees and POS tags) for extracting features. However, it has been shown that tweets are a very challenging domain for syntactic analysis, and parsing accuracies on Twitter are significantly lower than those on traditional text [Gimpel *et al.*, 2011; Kong *et al.*, 2014]. This limits the potentials of target-dependent SA systems that require external syntactic analyzers. On the other hand, automatic features based on distributed word representations has been shown to give competitive accuracies compared to manual features for SA of whole tweets [Tang *et al.*, 2014b; Kalchbrenner *et al.*, 2014]. In this paper, we explore a target-dependent Twitter SA model that does not use external syntactic analyzers, by leveraging distributed word representations and rich automatic features.

The architecture of the system is shown in Figure 1. Given a certain target in a tweet, we split the tweet into three components, including the target, its left context and its right context, assuming that the sentiment toward the target is decided by the interaction of both contexts. This context representation is independent of external syntactic analyzers. To model the interaction between the contexts and the target, we use word embeddings and neural pooling functions. Words in tweets are represented using two types of distributed word representations, including the skip-gram embeddings of Mik-

ilov et al. [2013] and the sentiment-driven embeddings of Tang et al. [2014b]. Neural pooling functions [Collobert *et al.*, 2011; Tang *et al.*, 2014b; Kalchbrenner *et al.*, 2014] are used to extract features automatically from both contexts according to each type of embeddings. In order to extract rich features, we explore a range of pooling functions, giving theoretical and empirical justifications.

Sentiment lexicons have been shown useful for SA [Wilson *et al.*, 2005; Hu and Liu, 2004], including SA on Twitter [Mohammad *et al.*, 2013]. However, no previous work has applied sentiment lexicons jointly with distributed word representations to improve SA. We follow the method of Moilanen et al. [2010] and exploit sentiment lexicons by filtering words in a tweet using them, resulting in new contexts for a given target, which contain sentiment-baring words only.

Though conceptually simple, the method is empirically highly effective. Experiments on a standard data set show that the proposed method outperforms the method of Dong et al. [2014] by 4.8% absolute accuracies, giving the best reported performance on the task.

The main contributions of this paper are three-fold:

- We propose a novel context representation for target-dependent Twitter SA, which is independent of syntactic analyzers, and incorporates sentiment lexicon information and distributed word representations;
- We explore a rich set of neural pooling functions for automatic feature extraction, drawing theoretical correlations behind these functions;
- We show that competitive results for target-dependent Twitter SA can be achieved using rich automatic features, reporting the best accuracies on a standard data set.

2 Related Work

SA on Twitter is pioneered by Go et al. [2009], who strictly follow Pang et al. [2002]'s seminal SA model. Their contribution is to leverage an emoticon-based tweet corpus as weakly labeled training data. Subsequent work tries to enrich the model by manually adding more complex features, such as proposing salience measures to discriminate common N-grams [Pak and Paroubek, 2010], extracting pattern-based features [Davidov *et al.*, 2010], building a tree kernel [Agarwal *et al.*, 2011] or using lexicon features [Kouloumpis *et al.*, 2011]. Mohammad et al. [2013] and Kiritchenko et al. [2014] build the state-of-the-art model by extracting rich manual features, including diverse sentiment lexicon (from general to specific) features and traditional features.

Twitter SA using word embeddings and automatic features has recently demonstrated large potentials. Tang et al. [2014b] and Tang et al. [2014a] demonstrate that automatic features can give the state-of-the-art accuracies on target-independent Twitter SA. They use neural network language models (NNLM) [Collobert *et al.*, 2011; Mikolov *et al.*, 2013] to automatically learn sentiment-driven embeddings, representing words in tweets with the embeddings, and use neural pooling functions to extract features automatically. Kalchbrenner et al. [2014] build a convolutional neural network with dynamic pooling functions to directly classify tweet opinions. Their method is extended by a character-level neural network to model morphological and shape information from words [dos Santos and Gatti, 2014]. We take the method of Tang et al. [2014b] as our target independent baseline.

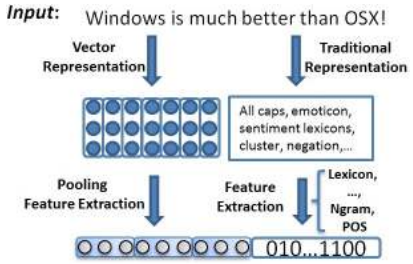

Figure 2: Feature extraction of Tang et al. [2014b].

For **target-dependent SA on tweets**, Jiang et al. [2011] analyze the dependency relationships between a given target and other words in a parse tree, considering them as additional features in the classical model of [Pang *et al.*, 2002]. Mitchell et al. [2013] extract the sentiment towards person and organization targets in a tweet by using CRFs and POS features. Dong et al. [2014] develop a novel deep learning approach based on automatically-parsed tweets, adaptively propagating sentiment signals to a specific target using a recursive neural network [Socher *et al.*, 2011]. To our knowledge, we are the first to exploit context-based patterns instead of syntax for targeted sentiment analysis on Twitter.

3 Baseline

Tang et al. [2014b] is a state-of-the-art model for target-independent SA on whole tweets. They use a linear model that takes a tweet as input, and outputs its sentiment polarity (i.e. negative or positive). As shown in Figure 2, the system represents a tweet using both distributed word representations (left) and traditional discrete word representations (right). From the vector representation, automatic features are extracted using a set of pooling functions. In addition to the *max* (maximum) function, which is the most widely used pooling function, Tang et al. [2014b] also use *min* (minimum) pooling and *avg* (average) pooling, obtaining a set of rich automatic features. From the discrete representation, traditional manual features, including lexicon features, n-gram features, and POS features, are extracted.

Tang et al. [2014b] empirically prove the power of the neural pooling functions in the un-targeted tweet SA task. We take the system of Tang et al. [2014b] as our baseline, but without using the discrete representation and traditional features. One contribution of Tang et al. [2014b] is that they use the sentiments of tweets as training data, and train a set of sentiment-baring word embeddings. They show that embeddings lead to significantly improved accuracies. We take their embeddings as a source of features, and additionally use standard embeddings and sentiment lexicons as other sources of features. Our baseline system give higher accuracies compared with Tang et al. [2014b].

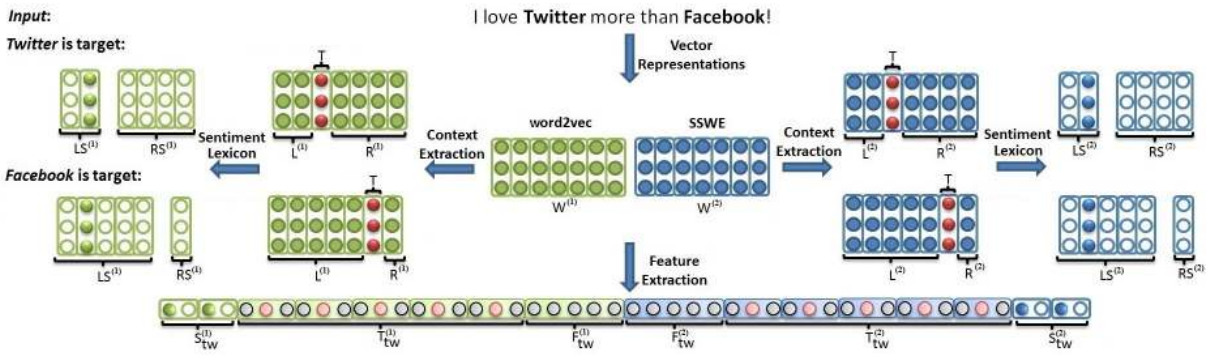

Figure 3: Feature extraction of our target-dependent system.

4 Method

As shown in Figure 1, our method consists of five main stages. Given a tweet, all its words are first mapped into distributed representations (4.1), before the left and right contexts of a given target are extracted (4.2). The full tweet, the left and right contexts and their lexicon-based alternatives (4.3) are used for feature extraction (4.4), and the resulting features are used as input for sentiment classification (4.5).

4.1 Distributed Word Representations

Word embeddings map words in a lexicon into low-dimensional vectors, with words having similar meanings being close to each other. Such representations have been shown effective in many NLP applications, for reducing data sparsity [Bengio *et al.*, 2003; Turian *et al.*, 2010; Collobert *et al.*, 2011] and learning sentence representations [Socher *et al.*, 2012; 2013; Kalchbrenner *et al.*, 2014].

We represent words in tweets using two types of word embeddings, in order to obtain rich sources of information for features. The first set of embeddings (*word2vec*) is trained using the skip-gram model of Mikolov et al. [2013], and the second set (*SSWE*) is trained using the sentiment-specific method of Tang et al. [2014b], which is a variation of Collobert et al. [2011]. In order to train embeddings with a larger-scale informal text corpus, we collect 5 million unlabeled tweets with more than 7 words through TwitterAPI[1].

Given a distributed word representation, a tweet can be represented as a matrix W with m rows and n columns, where m is the size of embeddings, and n is the length of the tweet. As shown in Figure 3, using the two types of word embeddings, two separate matrices are constructed for a given target.

4.2 Context Extraction

For target-dependent sentiments, we separate a tweet matrix W into three sub-matrices, L, T, and R, which correspond to the left context, the target, and the right context, respectively. The left and right contexts consist of all the words on the left and right of the target, respectively. The splitting of context is a salient difference between our target-dependent method (Figure 3) and the target-independent baseline (Figure 2). The sentiment towards a target results from the interaction between its left and right contexts.

As shown in Figure 3, for the tweet "I love Twitter more than Facebook!", when "Twitter" is a target, the left context L consists of word embeddings of "I love", and the right context R consists of word embeddings of "more than Facebook!". On the other hand, for the target "Facebook", the left context L consists of word embeddings of "I love Twitter more than", and the right context R consists of word embeddings of "!".

4.3 Lexicon-Based Distributed Contexts

Moilallen et al. [2010] show that a useful context for SA can be obtained by chunking a text according to sentiment polarities. We extend the contexts for a target using a similar method. Based on the two sub-matrices L and R, we generate two new context matrices LS and RS, by keeping embeddings of words in a sentiment lexicon, and filtering out the words that are not in the sentiment lexicon. Here, we do not care about the polarities of the words, only using the embedding context for feature extraction.

As shown in Figure 3, for the target "Twitter", only the embedding of the second word (i.e. "love") in the L matrix, which is contained in our sentiment lexicon, is kept, while the remaining word embeddings in L are replaced by zeros. The modified-matrices LS and RS contain prior sentiment knowledge in terms of its meaning and position.

4.4 Feature Extraction

Based on the rich set of contexts from both embeddings, row-wise pooling functions are performed in order to automatically extract features.

Source of Features

- **Target-dependent features from *L*, *R*, and *T*:** As shown in Figure 3, target-dependent features (i.e. T_{tw}) consists of features from the left context L, the given target T and the right context R. When *word2vec* embeddings are used, the resulting features are

$$T_{tw}^{(1)} = [P(L^{(1)}), P(T^{(1)}), P(R^{(1)})], \quad (1)$$

and when *SSWE* is used, the resulting features are

$$T_{tw}^{(2)} = [P(L^{(2)}), P(T^{(2)}), P(R^{(2)})], \quad (2)$$

[1] https://twitter.com/twitterapi.

where $P(X)$ represents the feature extraction function, which consists of k different pooling functions f_p, (p $\in$ [1,..., k]) on an embedding matrix X

$$P(X) = [f_1(X), ..., f_k(X)] \quad (3)$$

- **Target-dependent features from *LS*, *RS*:** Similar to T_{tw}, we extract sentiment-baring features (i.e. S_{tw}) from the lexicon-based contexts LS and RS. Using the *word2vec* embeddings, the resulting features are

$$S_{tw}^{(1)} = [P(LS^{(1)}), P(RS^{(1)})], \quad (4)$$

and using *SSWE*, the resulting features are

$$S_{tw}^{(2)} = [P(LS^{(2)}), P(RS^{(2)})] \quad (5)$$

- **Full tweet features:** We additionally extract features from the full tweet (i.e. F_{tw}), in order to model interactions between the two contexts. The features extracted on *word2vec* and *SSWE* embeddings are

$$F_{tw}^{(1)} = P(W^{(1)}), \quad (6)$$

and

$$F_{tw}^{(2)} = P(W^{(2)}), \quad (7)$$

respectively, where $W^{(1)}$ and $W^{(2)}$ are the embedding matrices of a given tweet using *word2vec* and *SSWE*, respectively.

Finally, $F_{tw}^{(1)}, T_{tw}^{(1)}, S_{tw}^{(1)}, F_{tw}^{(2)}, T_{tw}^{(2)}$ and $S_{tw}^{(2)}$ are concatenated into a final feature vector P_{tw} for three-way sentiment classification of the target.

$$P_{tw} = [F_{tw}^{(1)}, T_{tw}^{(1)}, S_{tw}^{(1)}, F_{tw}^{(2)}, T_{tw}^{(2)}, S_{tw}^{(2)}] \quad (8)$$

Neural Pooling Functions

Pooling functions have been shown highly effective for feature selection from dense real-valued feature vectors [Collobert *et al.*, 2011; Socher *et al.*, 2011; Tang *et al.*, 2014b]. Given n vectors of size m, dimension-wise max pooling have been commonly used in neural network models for feature extraction, resulting in a m-dimensional dense feature vector. Tang et al. [2014b] applied the max, min, and avg pooling functions to extract features from word embeddings for a linear model. Though empirically highly useful, the reason behind the effectiveness of the min and avg are not justified by Tang et al. [2014b]. We try to give some intuitive and theoretical justifications to these functions, while proposing more rich functions for pooling.

Intuitively, m-dimensional vectors contain m automatic features about words. max pooling selects the highest values of each feature among n words (e.g. most positive sentiment), and min pooling selects the lowest values (e.g. most negative sentiment). avg is a combination of feature values in each dimension (e.g. averaged sentiment). The three functions collect different statistics of the n words, and can be seen as instances of a generalized norm function, which combines features dimension-wise

$$\overline{x_i} = (\sum_{k=1}^{n} x_{i,k}^{p})^{\frac{1}{p}} \quad (9)$$

Where $x_{i,k}$ is the ith feature of the k word. The max function corresponds to Equation 9 with $p = \infty$, the min function corresponds to Equation 9 with $p = -\infty$ and the avg function corresponds to Equation 9 with $p = 1$. We further generalize this idea of feature extraction via the norm function, and propose additional pooling functions.

In particular, when the original coordinates of the feature space shifts from $\overrightarrow{0}$ to the average of all vectors, the standard deviation (std) corresponds to Equation 9 with $p = 2$. Intuitively, std represents sentiment variation. When the feature space shift to the power space, the product pro corresponds to Equation 9 with $p = 1$. Intuitively, pro is a variation of avg, with larger contrast between positive and negative values.

4.5 Sentiment Classification

The input to the final sentiment classifier is the set of rich real-values features P_{tw}, and the output is the sentiment class $S \in \{-1, 0, +1\}$. We follow the state-of-the-art approaches [Mohammad *et al.*, 2013; Kiritchenko *et al.*, 2014; Tang *et al.*, 2014b] and use LibLinear[2], which is widely used for classification with large and dense features [Fan *et al.*, 2008].

For binary classification, given a set of feature-label pairs $(x_i, y_i), i = 1, ..., l, x_i \in R^n, y_i \in \{-1, +1\}$, the linear model is trained by optimizing the objective function:

$$\min_{\omega} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l} L(\omega; x_i, y_i) \quad (10)$$

where C > 0 is a penalty parameter, and $L(\omega; x_i, y_i)$ is a loss function.

During testing, given the feature vector of an unlabeled example x, the linear model performs classification by using the decision function:

$$y = sign \ \ (\omega^T \cdot x), \qquad (y \in \{-1, +1\}) \quad (11)$$

For three-way classification, we apply the one-vs-the-rest strategy. For training, the L2 loss function is used for L, and the C parameter is tuned by cross-validation.

5 Experiments

We perform a set of development experiments to evaluate the effectiveness of embeddings, context patterns, pooling functions, and sentiment lexicons on the performance of the proposed approach, tuning parameter values for our final model. A final test is performed under the best development settings in order to evaluate the model in comparison with previous work.

5.1 Experimental Data and Settings

Data sets: Our experiments are carried out on the target-dependent data set of Dong et al. [2014], which is manually annotated with sentiment labels (negative, positive, and neutral) toward given targets (such as "bill gates", "google" and "xbox"). The data set includes 6248 training tweets and 692 testing tweets, with a balanced number of positive, negative,

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

Sentiment Lexicon	Positive	Negative
MPQA	2289	4114
HL	2003	4780
NRC	2231	3243
MRQA⊕HL	2706	5069
MRQA⊕HL⊕NRC	3940	6490

Table 1: Statistics of sentiment lexicons.

and neutral tweets (25%, 25%, and 50%, respectively). Dong et al. [2014] report that the data set is annotated with 82.5% agreement between human annotators.

We apply TwitterAPI to crawl tweets containing one of following emoticons: ":), :), : −), : (, : (, : −(". Downloaded tweets are tokenized using TwitterNLP [Gimpel *et al.*, 2011], and filtered if they contain less than 7 tokens. Eventually, we collect around 5 million tweets for learning word embeddings using the continuous skip-gram model. We use the *SSWE* data[3] to obtain *SSWE* embeddings.

We use three sentiment lexicons, namely MPQA[4] [Wilson *et al.*, 2005], HL[5] [Hu and Liu, 2004], and NRC emotion lexicon[6] [Mohammad and Yang, 2011], integrating them to filter the context. Table 1 presents the statistics of the three sentiment lexicons. To combine these sentiment lexicons (⊕), we compute the union between them, and filter at words baring both positive and negative sentiments.

Experimental settings: To learn distributed word representations using the *word2vec* package[7], we empirically choose 100, 3, and 10 for the embedding size, window length, and word count threshold, respectively. For tuning of a final three-way classification model, we perform five-fold cross validation on the training data to adjust features and the penalty parameter C. Then, the model with the best features and optimal C value is applied on the testing set.

Evaluation metrics: We follow previous work [Jiang *et al.*, 2011; Dong *et al.*, 2014; Tang *et al.*, 2014b] and use the Accuracy of three-way classification for each target as the main evaluation metric. In the final test, we additionally take the macro-average F1-score over the three classes for comparison with other methods.

5.2 Models

To examine the contributions of various features to target-dependent SA on Twitter, we build the following models:

Target-ind: Our baseline target independent approach, which uses full tweet features (F_{tw});

Target-dep$^-$: A pure targeted method, which makes use of target-dependent features (T_{tw});

Target-dep: This method combines features of both **Target-ind** and **Target-dep$^-$**;

[3]http://ir.hit.edu.cn/~dytang/

[4]http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/.

[5]http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

[6]http://saifmohammad.com/WebPages/ResearchInterests.html\#SentimentAnalysis.

[7]https://code.google.com/p/word2vec/.

Pooling function	CV Acc
max	63.46
$max + min$	64.72
$max + avg$	65.26
$max + min + avg$	65.59
$max + min + avg + std$	65.61
$max + min + avg + pro$	65.62
$\boldsymbol{max + min + avg + std + pro}$	**65.72**

Table 2: Effectiveness of Pooling Functions.

Feature Types	C Value	CV Acc
Target-ind	0.12	59.22
Target-dep$^-$	0.025	65.38
Target-dep	**0.01**	**65.72**

Table 3: Target-independent SA vs target-dependent SA.

Target-dep$^+$: The method uses **Target-dep** features and target-dependent sentiment features (S_{tw}). This model is our final model described in Section 4.

5.3 Development Experiments

We start with a vanilla **Target-dep** model, and study the effectiveness of pooling functions, contexts, lexicons and embeddings by incrementally tuning each setting based on the previous optimal configuration.

Effectiveness of Pooling Functions

We survey the contribution of various pooling functions by using the **Target-dep** model and *word2vec* on the cross-validation data. Table 2 shows the results. With the use of max pooling alone to extract features, the accuracy is 63.46%. By using combined max, min and avg pooling, the model gives a significantly improved accuracy of 65.59%. This observation is consistent with Tang et al. [2014b], showing that the additional pooling functions are highly effective also for target-dependent Twitter SA. Both std and pro further enhance the performance. By combining all the five pooling functions, we obtain the best accuracy (65.72%), and we apply the model to the next experiments.

Effectiveness of Target-dependent Features

We assess the effectiveness of our rich target-dependent features by comparing the **Target-ind**, **Target-dep$^-$** and **Target-dep** models on the cross-validation data. *word2vec* embeddings are used by all the three models.

The results are shown in Table 3. **Target-ind** gives an accuracy of 59.22%. Using target-dependent features from the left context (*L*), the given target (*T*), and the right context (*R*), **Target-dep$^-$** gives a significantly better accuracy of 65.38%, 6.16% higher than the **Target-ind** baseline. **Target-dep** further captures the interaction between *L* and *R* by using features over the whole tweet, and gives the best accuracy of 65.72%. This experiment shows that target-dependent contexts play a vital role in classifying targeted tweet sentiment, and automatic features can capture the influence of context patterns on the sentiment.

Sentiment Lexicon	CV Acc
Target-dep	65.72
Target-dep$^+$: NRC	66.05
Target-dep$^+$: HL	67.24
Target-dep$^+$: MPQA	65.56
Target-dep$^+$: MPQA⊕HL	**67.40**
Target-dep$^+$: MPQA⊕HL⊕NRC	67.30

Table 4: Effectiveness of Sentiment Lexicons.

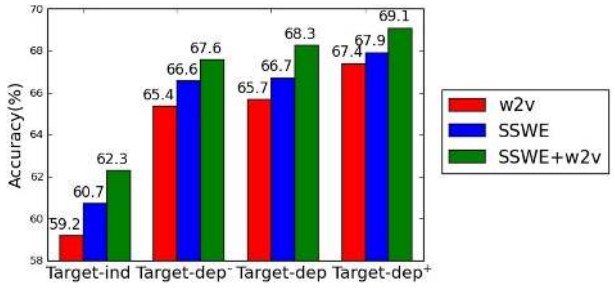

Figure 4: Effectiveness of word embedding.

Effectiveness of Sentiment Lexicons

We exploit the contributions of the sentiment lexicons in Table 1 to our method. Lexicon contexts are extracted and added into the **Target-dep** model to obtain the **Target-dep**$^+$ model. The results are obtained using *word2vec* embeddings on the cross-validation data.

As shown in Table 4, before adding sentiment features, the **Target-dep** approach gives an accuracy of 65.72%. We subsequently add the NRC, HL, and MPQA sentiment lexicons into the model. Except for the MPQA lexicon, which does not have a significant impact, the remaining lexicons give higher accuracies, with HL giving the most improvement (67.24%). In addition, by combining MPQA and HL, **Target-dep**$^+$ reaches the best result (67.40%). We apply this best model to the rest of our experiments.

Effectiveness of Word Embeddings

The previous development experiments are all based on *word2vec* embeddings only. We further study the effect of word embedding models by comparing the performance of our methods (i.e. **Target-ind**, **Target-dep**$^-$, **Target-dep** and **Target-dep**$^+$) under the best previous settings, by contrasting and combining the *word2vec* and *SSWE* embeddings. The results are obtained on the cross-validation data.

As illustrated in Figure 4, the models using *SSWE* embeddings yield better performance compared to those using *word2vec* embeddings. This is likely because *SSWE* contains more sentiment signals from the emoticon-based data. The accuracies of our models using both embeddings are consistently higher than those using individual embeddings. By using both embeddings in **Target-dep**$^+$, our method achieves the best results (69.1%).

5.4 Final Results

We evaluate the performance of **Target-ind**, **Target-dep** and **Target-dep**$^+$ on the test data, with the best configurations obtained in cross-validation tuning. The results are compared with the following models:

Method	Acc	F1
SSWE [Tang *et al.*, 2014b]	62.4	60.5
Target-ind	**67.3**	**66.4**
SVM-dep [Jiang *et al.*, 2011]	63.4	63.3
AdaRNN [Dong *et al.*, 2014]	66.3	65.9
Target-dep	**69.7**	**68.0**
Target-dep$^+$	**71.1**	**69.9**

Table 5: Evaluation results.

***SSWE*:** the target-independent method of Tang et al. [2014b]. Liblinear is used for three-way sentiment classification.

SVM-dep: the target-dependent method of Jiang et al. [2011]. Liblinear is used for three-way sentiment classification.

AdaRNN: the method of Dong et al. [2014]. SVM is used for three-way classification.

The results are shown in Table 5. As a target-independent SA baseline, *SSWE* gives a 62.4% accuracy of three-way targeted classification. **Target-ind** can be viewed as an alternative version of *SSWE*, without using the traditional ngram and POS features, but defining richer automatic features by using the embeddings of Mikolov et al. [2013] in addition the those of Tang et al. [2014b], and applying more pooling functions (std and pro). The number of features in **Target-ind** roughly triples that in *SSWE*. It gives a significantly higher accuracy compared to *SSWE*, showing the effectiveness of rich sources of automatic features.

Among targeted approaches, Jiang et al. [2011] gives 1% higher accuracy compared with *SSWE*, thanks to target-dependent features. Using a recursive neural network on a parse tree, *AdaRNN* gives an accuracy of 66.3%, which is the previous best result on the data set. By the use of automatic rich features, **Target-dep** gives a accuracy of 69.7%, higher than that of *AdaRNN*. The results demonstrate that competitive accuracies can be obtained without the use of syntactic analyzers. Eventually, with the use of sentiment lexicons in the embedding space, the accuracy of our system, **Target-dep**$^+$, reaches 71.1%, achieving the current best result.

6 Conclusion

We studied target-dependent Twitter sentiment classification by making use of rich automatic features based on distributed word representations. Our method, which is independent of external syntactic analyzers, gives better performance compared to the best previous method that uses syntax. The method solves the potential limitation of syntax-based method by avoiding the influence of noise by automatic syntactic analyzer. Our experiments show that multiple embeddings, multiple pooling functions and sentiment lexicons offer rich sources of feature information, which leads to significant improvement on accuracies.

References

[Agarwal *et al.*, 2011] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proc. LSM*, 2011.

[Bengio *et al.*, 2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12, November 2011.

[Davidov *et al.*, 2010] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proc. CoNLL*, 2010.

[Dong *et al.*, 2014] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proc. ACL*, 2014.

[dos Santos and Gatti, 2014] Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proc. COLING*, 2014.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[Gimpel *et al.*, 2011] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and A. Noah Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. ACL HLT*, 2011.

[Go *et al.*, 2009] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

[Hu and Liu, 2004] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proc. KDD*, 2004.

[Jiang *et al.*, 2011] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proc. ACL HLT*, 2011.

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proc. ACL*, 2014.

[Kiritchenko *et al.*, 2014] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *J. Artificial Intelligence Research*, 2014.

[Kong *et al.*, 2014] Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and A. Noah Smith. A dependency parser for tweets. In *Proc. EMNLP*, 2014.

[Kouloumpis *et al.*, 2011] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! *Proc. ICWSM*, 2011.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, 2013.

[Mitchell *et al.*, 2013] Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. Open domain targeted sentiment. In *Proc. EMNLP*, 2013.

[Mohammad and Yang, 2011] Saif Mohammad and Tony Yang. Tracking sentiment in mail: How genders differ on emotional axes. In *Proc. WASSA*, 2011.

[Mohammad *et al.*, 2013] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proc. SemEval*, 2013.

[Moilanen *et al.*, 2010] Karo Moilanen, Stephen Pulman, and Yue Zhang. Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression. In *Proc. WASSA*, 2010.

[Pak and Paroubek, 2010] Alexander Pak and Patrick Paroubek. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In *Proc. SemEval*, 2010.

[Pang *et al.*, 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. EMNLP*, 2002.

[Socher *et al.*, 2011] Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proc. ICML*, 2011.

[Socher *et al.*, 2012] Richard Socher, Brody Huval, D. Christopher Manning, and Y. Andrew Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proc. EMPNL/CoNLL*, 2012.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, D. Christopher Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP*, 2013.

[Tang *et al.*, 2014a] Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proc. COLING*, 2014.

[Tang *et al.*, 2014b] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proc. ACL*, 2014.

[Turian *et al.*, 2010] Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proc. ACL*, 2010.

[Wilson *et al.*, 2005] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. EMNLP HLT*, 2005.