Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Improving Twitter Sentiment Classification Using Topic-Enriched Multi-Prototype Word Embeddings

Yafeng Ren[1], **Yue Zhang**[2], **Meishan Zhang**[3*] **and Donghong Ji**[1*]
1. Computer School, Wuhan University, Wuhan, China
2. Singapore University of Technology and Design, Singapore
3. School of Computer Science and Technology, Heilongjiang University, Harbin, China
{renyafeng, dhji}@whu.edu.cn
{yue_zhang, meishan_zhang}@sutd.edu.sg

Abstract

It has been shown that learning distributed word representations is highly useful for Twitter sentiment classification. Most existing models rely on a single distributed representation for each word. This is problematic for sentiment classification because words are often polysemous and each word can contain different sentiment polarities under different topics. We address this issue by learning topic-enriched multi-prototype word embeddings (TMWE). In particular, we develop two neural networks which 1) learn word embeddings that better capture tweet context by incorporating topic information, and 2) learn topic-enriched multiple prototype embeddings for each word. Experiments on Twitter sentiment benchmark datasets in SemEval 2013 show that TMWE outperforms the top system with hand-crafted features, and the current best neural network model.

Introduction

Twitter sentiment classification, which classifies the sentiment polarity of a tweet as positive, neutral or negative, has received much research attention in recent years (Jiang et al. 2011; Liu 2012; Hu et al. 2013; Tang et al. 2014). Most existing work follows Pang et al. (2002) and Go et al. (2009), treating Twitter sentiment classification as a special case of text classification, and investigating the use of effective features (Pang and Lee 2008; Maas et al. 2011; Owoputi et al. 2012; Labutov and Lipson 2013). In particular, Mohammad et al. (2013) build the top-performing system in the Twitter sentiment classification track of SemEval 2013 (Nakov et al. 2013), using diverse sentiment lexicons and a variety of hand-crafted features. Tang et al. (2014) propose sentiment-specific word embedding, which gives better performance compared with the current best system using diverse word representations.

Existing word embedding models for Twitter sentiment classification, however, have two limitations. On the one hand, one embedding is generated for each word, despite that sentiment-baring words may be polysemous. On the other hand, most methods ignore topic information of the tweet in the word embedding space, which determines the meaning of polysemous words. For example, two tweets from the SemEval 2013 dataset are given below, with the word "offensive" appearing in different sentiment polarities.

- Monday before I leave Singapore, I am going to post something that might be **offensive**. (**NEGATIVE**)
- #Patriots Tom Brady wins AFC **offensive** player of the week for 22nd time.... http://t.co/WlFHyQ0I – #NFL (**POSITIVE**)

The same opinion expression "offensive" demonstrates different sentiment polarities in the above two tweets, because the topics are different.

In this paper, we show that significant improvements can be achieved by using topic-enriched multi-prototype word embeddings. First, we build a baseline system using neural network, which has been shown to give competitive results compared with traditional linear models (Socher et al. 2013; dos Santos and Gatti 2014). The baseline model learns word embedding features by incorporating both local (n-gram) and global context (sentiment and topic distribution). It gives comparable results to the best models in the literature. Second, we extend the baselines system to learn multiple embeddings for each word, which gives better performance. Finally, performance can be further improved by integrating our proposed model and the current best model using diverse word representations.

Compared with the baseline, the final system improves two-category classification performance on the standard Semeval 2013 dataset from 82.17% to 87.97% in macro-F score. The result is significantly higher compared with the best results from the SemEval 2013 participants (84.73%) and the best reported result on the same dataset (84.98%).

Related Work

Existing methods on Twitter sentiment analysis focus on feature engineering. The performance of a sentiment classifier is heavily dependent on the choice of feature representations. Pang et al. (2002) pioneer this field by using bag-of-word feature representations. Many different feature learning methods are proposed subsequently to obtain better performance (Pang and Lee 2008; Liu 2012; Owoputi et al. 2012; Feldman 2013; Mohammad, Kiritchenko, and Zhu 2013).

Recent studies investigate learning low-dimensional, dense and real-valued word embedding vectors for senti-

*corresponding author

ment classification. Maas et al. (2011) propose a probabilistic document model following Blei et al. (2003). Labutov and Lipson (2013) re-embed words from existing word embeddings. Tang et al. (2014) develop three neural network models to learn word vectors from tweets containing positive and negative emoticons. These methods have shown promising results when combined with linear or neural classifiers (Tang et al. 2014; Vanzo, Croce, and Basili 2014).

Our work is also related to learn multi-prototype word embeddings. Huang et al. (2012) propose to leverage global contextual information and multi-prototype embeddings to achieve performance gains on word-similarity tasks. Tian et al. (2014) propose to learn multiple embedding vectors for polysemous words from a probabilistic perspective, by designing an Expectation-Maximization algorithm. However, for Twitter sentiment classification, existing work focuses on learning single-prototype word embeddings. We propose to learn topic-enriched multi-prototype word embeddings for Twitter sentiment classification, using the embedding framework of Collobert et al. (2011) and Tang et al. (2014).

Learning Multi-Prototype Word Embeddings for Twitter Sentiment Classification

We propose to learn topic-sensitive representations for words by integrating topic information, using the algorithm of Collobert et al. (2011) to learn different types of embeddings. Two embedding models are developed. First, the single-prototype embedding approach of Collobert et al. (2011) is extended by incorporating topic information, demonstrating the influence of such information on Twitter sentiment analysis. Second, the embedding method is further extended by allowing multiple prototypes for each word, fully leveraging topic information in each prototype. Following Tang et al. (2014), we also incorporate sentiment information into a word.

C&W Model

Collobert et al. (2011) build a neural model to learn word embeddings from n-gram contexts. When training embedding for the word w^c using n-grams, w^c is replaced with random word w^r to form corrupted n-grams. The training objective is that the score of original n-gram should be larger than the score of corrupted n-gram by a margin of 1 according to a neural network. This ranking objective function can be optimized by using a hinge loss,

$$loss_{syn}(t^c) = max(0, 1 - f^{syn}(t^c) + f^{syn}(t^r)) \tag{1}$$

where t^c is the original n-gram, t^r is the corrupted n-gram and $f^{syn}(\cdot)$ represents the n-gram score computed by the neural network.

Figure 1(a) shows the C&W model using a simple three-layer neural network. From the bottom, the first layer is a lookup layer, where the representation L_t of each word w_t is looked up in a dictionary L. Then a linear transformation is applied to the concatenation of all the word embeddings in a fixed-size window. Assuming that the word embedding size is s, the length of hidden layer is h and the window size is m, a *hTanh* layer is used on top of the linear transformation layer to obtain,

$$a = h_1(W_1 * [L_1; L_2; \cdots; L_m] + b_1) \tag{2}$$

where $W_1 \in R^{h\times(s*m)}, b_1 \in R^h$, and $h_1(\cdot)$ is the activation function. The final n-gram score is computed as:

$$f^{syn}(t) = W_2 * a \tag{3}$$

where $W_2 \in R^{1\times h}$. No bias term is used here.

After the scores for both the original and the corrupted n-grams are obtained, the cost by the hinge loss function can be computed.

Topic-Enriched Word Embeddings

The C&W model captures the semantic relation between words in local contexts, which is proved useful in many NLP tasks (Turian, Ratinov, and Bengio 2010; Zhang and Zhang 2015), but is not directly oriented for sentiment classification. To address the problem, Tang et al. (2014) propose to modify the original C&W model, using the sentiment polarity of a tweet to augment the loss function (SSWE), which is shown in Figure 1(b). SSWE demonstrates the effectiveness of sentiment information in word embeddings for Twitter sentiment classification. Following the neural network structure of Tang et al. (2014), we further integrate local (n-gram) and global context (sentiment, topic distribution) to learn word embeddings. Topic information has been shown useful for Twitter sentiment classification (Xiang and Zhou 2014). We develop two basic models (i.e. TEWE, TSWE) by integrating different global information into the neural networks.

Model 1 (TEWE) An intuitive solution to integrate topic information is predicting the topic distribution of text based on input n-grams. Assuming that there are M topics, we add a hidden layer of M nodes under the top $softmax$ layer, and additionally predict the topic in the output layer. The neural network is given in Figure 1(c). Unlike C&W, our embedding (Topic-Enriched Word Embedding, TEWE) does not generate any corrupted n-grams. Let $g^{top}(t)$ be the gold M-dimensional multinomial distribution of input t ($\sum_{j=1}^{M} g_j^{top}(t) = 1$). Here, the topic distribution $g^{top}(t)$ is generated using LDA (Blei et al., 2003). The cross-entropy loss of the *softmax* layer is shown in Equation (4):

$$\begin{aligned} f^{top}(t) &= softmax(W_t * a + b_t) \\ loss_{top} &= -\sum_{i=1}^{M} f_i^{top}(t) * log(g_i^{top}(t)) \end{aligned} \tag{4}$$

Model 2 (TSWE) The sentiment (SSWE) and topic (TEWE) of a Tweet are related semantically. In Figure 1(d), we build a neural network, incorporating both sentiment and topic information into the C&W model. We add two *softmax* output layers, one for sentiment distribution and the other for topic distribution, as introduced in formula (4) and (5), respectively. Let $g^{sen}(t)$ be the gold K-dimensional multinomial distribution of input *t*, and $\sum_{k=1}^{K} g_k^{sen}(t) = 1$, where

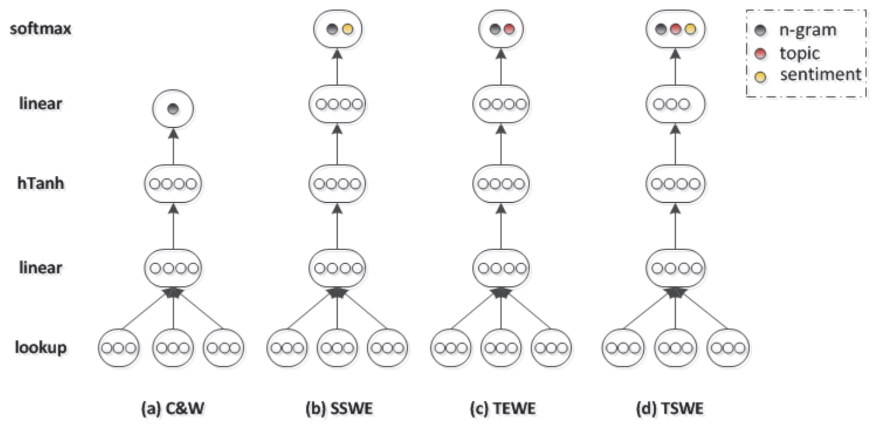

Figure 1: The C&W, SSWE models and our neural networks (TEWE and TSWE) for learning word embeddings.

K denotes the number of sentiment categories. The cross-entropy loss of the *softmax* layer is:

$$f^{sen}(t) = softmax(W_s * a + b_s)$$
$$loss_{sen} = -\sum_{j=1}^{K} f_j^{sen}(t) * log(g_j^{sen}(t)) \tag{5}$$

where $g^{sen}(t)$ represents the gold sentiment label.

In addition, we also capture n-gram information. The total loss of Topic and Sentiment-Enriched Word Embedding (TSWE) is a mixture of n-gram, topic and sentiment loss:

$$J = \frac{1}{N}\sum_t [\alpha * loss_{sen}(t) + \beta * loss_{top}(t) + (1-\alpha-\beta) * loss_{syn}(t)] \tag{6}$$

where α and β are linear interpolation weights.

Model Training To train TEWE and TSWE, we crawl 10M tweets, which include 5M tweets with positive emoticons and 5M with negative emoticons. We train TEWE and TSWE by taking the derivatives of the loss through back-propagation with respect to the whole set of parameters (Collobert et al. 2011), and using AdaGrad (Duchi, Hazan, and Singer 2011) to update the parameters. We empirically set the window size as 3, the embedding length as 50, the length of hidden layer as 100, the activation functions $h1$ as $tanh$, and the initial learning rate of AdaGrad as 0.1.

Multi-Prototype Word Embeddings

Topic-enriched embeddings (TEWE and TSWE) incorporate non-local tweet context into distributed word representations, thereby improving the performance of sentiment classification. However, they do not fully exploit the advantage of topical context, because polysemous words have the same embeddings under varying contexts.

In order to learn multiple prototype embeddings, we first identify polysemous words, by simply using a frequency threshold, which is set three times of the prototype number *p*. For each instance of high-frequency words, we calculate an "environment vector" *Env*, which consists of *idf*-weighted average word embeddings in the context, and the topic information:

$$Env = [\sum_{i \in context} L_i * idf_{w_i}; g^{top}] \tag{7}$$

Here, L_i denotes the embedding of w_i in the context, which is consistent with Equation 2. The context is the word embeddings computed in the TSWE model over the tweet. After obtaining *Env* for all the instances of a word, we conduct k-means clustering on the vectors, where the distance is defined as the cosine distance:

$$d(Env_1, Env_2) = 1 - \frac{Env_1 * Env_2}{||Env_1|| \cdot ||Env_2||} \tag{8}$$

In our model, *p* is set to 10. Based on the k-means results, the model generates 10 initial prototypes for high-frequency words. We then reduce the number of prototypes for some words by merging clusters when the distances between the clustering centers are sufficiently close (i.e. $\leq \delta$). After this step, a cluster is regarded as a prototype. The final center of each cluster is recorded.

Finally, the embedding dictionary is extended to contain different cluster centers of the same words, and the corpus is relabeled with the new dictionary. The new corpus and dictionary are used in the TEWE and TSWE models to train word embeddings with multiple prototypes (M-TEWE and M-TSWE), respectively. We can generate 6-8 prototypes for each word on average.

Sentiment Classification Model

We conduct sentiment classification using a convolutional neural network. Shown in Figure 2, it consists of five layers, including an input layer, a convolutional layer, a pooling layer, a non-linear hidden layer for feature combination and an output layer.

Input layer

Nodes of the input layer denote words in the tweet in their written order. For each word w_i, we first compute the *Env* vector based on its context and the topic distribution of the sentence. Then we compare the *Env* vector with all cluster

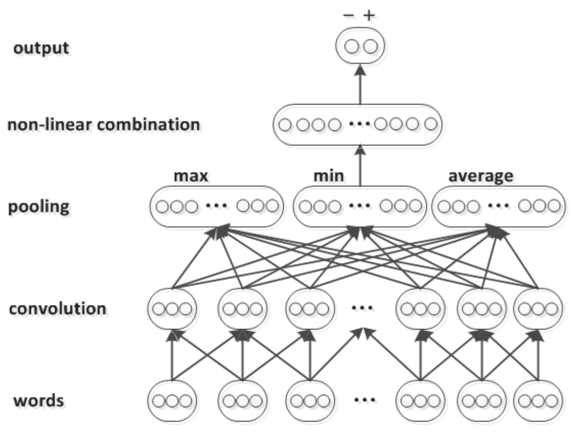

Figure 2: Convolutional neural network for Twitter sentiment classification.

centers of multiple word prototypes, and find the best prototype for the word w_i. We use a look-up matrix $\mathbf{E}$ to obtain its embedding $e(w_i) \in R^{\mathrm{D}\times 1}$, where $\mathbf{E} \in R^{\mathrm{D}\times V}$ is a model parameter, *D* is the word vector dimension and *V* is the vocabulary size. In this paper, **E** is obtained via the TEWE or TSWE model over raw corpus.

Convolution layer

The convolution action has been commonly used in neural networks to synthesize lexical n-gram information (Collobert et al. 2011). Because n-grams have been shown useful for Twitter sentiment analysis (Mohammad, Kiritchenko, and Zhu 2013)(dos Santos and Gatti 2014), we apply them to our neural network. Given the input layer node sequence $e(w_1)\cdots e(w_n)$, we use the convolution operation to obtain a new hidden node sequence $\vec{h}^1_1\cdots\vec{h}^1_n$ according to the following equation:

$$\vec{h}^1_i = \mathit{tanh}(\mathbf{W}^1 \cdot [e'(w_{i-1}), e'(w_i), e'(w_{i+1}), 1]'),$$

where e' denotes the transpose of the vector e, $\mathbf{W}^1 \in R^{\mathrm{C}\times(3\mathrm{D}+1)}$ is a model parameter, and C is the output dimension. We consider a window size of 3 to combine word embedding features, and use *tanh* as the activation function for this hidden layer.

Pooling layer

We exploit pooling techniques to merge the varying number of features from the convolution layer into vectors with fixed dimensions. The most common pooling technique is the *max* function, which chooses the highest value on each dimension from a set of vectors. On the other hand, *min* and *average* pooling have also been used for sentiment classification (Tang et al. 2014; Vo and Zhang 2015), giving significant improvements. We consider all the three pooling methods, concatenating them as a new hidden layer $\vec{h}^2$. Formally, the values of $\vec{h}^2$ is defined as:

$$\vec{h}^2 = \Big[\begin{bmatrix} \max(\vec{h}^1_{i1}) \\ \cdots \\ \max(\vec{h}^1_{iC}) \end{bmatrix}', \begin{bmatrix} \min(\vec{h}^1_{i1}) \\ \cdots \\ \min(\vec{h}^1_{iC}) \end{bmatrix}', \begin{bmatrix} avg(\vec{h}^1_{i1}) \\ \cdots \\ avg(\vec{h}^1_{iC}) \end{bmatrix}' \Big],$$

where $\vec{h}^1_{ij}$ denote the jth dimension of $\vec{h}^1_i$.

Hidden layer

We use a non-linear hidden layer to automatically combine the *max*, *min* and *average* pooling features. The value of the hidden layer can be computed as:

$$\vec{h}^3 = \mathit{tanh}(\mathbf{W}^2 \cdot \begin{bmatrix} \vec{h}^2 \\ 1 \end{bmatrix}),$$

where $\mathbf{W}^2 \in R^{\mathrm{H}\times(3\mathrm{C}+1)}$ is a model parameter, and H is the size of this layer.

Output layer

Finally, an output layer is applied to score all possible labels according to the features in the hidden layer. The output is computed using a linear transformation:

$$\vec{o} = \mathbf{W}^{\mathrm{o}} \cdot \vec{h}^3,$$

where $\mathbf{W}^{\mathrm{o}} \in R^{2\times\mathrm{H}}$ is the model parameter for output layer.

Training

Our training objective is to minimize the cross-entropy loss over a set of training examples $(x_i, y_i)|_{i=1}^N$, plus a l_2-regularization term,

$$L(\theta) = -\sum_{i=1}^{N} \log \frac{e^{\vec{o}(y_i)}}{e^{\vec{o}(0)} + e^{\vec{o}(1)}} + \frac{\lambda}{2} \parallel \theta \parallel^2,$$

where θ is the set of model parameters, including $\mathbf{W}^1$, $\mathbf{W}^2$ and $\mathbf{W}^{\mathrm{o}}$. We use online AdaGrad (Duchi, Hazan, and Singer 2011) to minimize the objective function, with an initial learning rate of 0.01. We follow Glorot et al. (2010) and initialize all the matrix and vector parameters with uniform samples in $(-\sqrt{6/(r+c)}, -\sqrt{6/(r+c)})$, where r and c are the numbers of rows and columns of the matrixes, respectively.

Experiments

We evaluate our proposed embeddings by incorporating them into the neural classifier for Twitter sentiment analysis. We also evaluate the effectiveness of TSWE by measuring word similarity in the embedding space for sentiment lexicons.

Dataset and Evaluation

We conduct experiments on the Twitter sentiment classification benchmark in SemEval 2013 (Nakov et al. 2013). The training and development sets cannot be fully downloaded, because some tweets were not available due to modified authorization status[1]. The test set is directly provided to the participants. The distribution of our dataset is given in Table 1. We train the CNN sentiment classifier on the training set, tune parameters on the development set and evaluate the final model on the test set. The classification results are measured by macro-F scores.

[1]we downloaded the tweets on 10/17/2014

	Positive	Negative	Neutral	Total
Train	2,631	986	3,420	7,037
Dev	397	208	482	1,087
Test	1,534	587	1,614	3,735

Table 1: Statistics of the SemEval 2013 Twitter sentiment classification dataset.

Type	#parameter
Network structure	D = 50, C = 100, H = 50
Training	$\lambda = 10^{-8}$, $\eta = 0.01$

Table 2: Hyper-parameter values in the final model.

	Methods	Macro-F score (%)
SINGLE	C&W	67.75
	TEWE	82.17
	SSWE (The Current Best)	84.98
	TSWE	**85.34**
MULTI	M-C&W	72.37
	M-TEWE	83.64
	M-SSWE	86.10
	M-TSWE	**86.83**
	NRC (Top System in SemEval)	84.73
	M-TSWE+NRC	**87.97**

Table 3: Macro-F score based on different word embeddings.

Hyper-Parameters

There are five hyper-parameters in the final model, including the network structure parameters (i.e. the sizes of word vectors *D* and the output dimensions of the word convolution layer *C*, the output dimension of the non-linear combination layer *H*), the parameters for supervised training (i.e. the l_2-regularization co-efficient λ and initial learning rate η for AdaGrad). The values of the parameters are set according to common values in the machine learning literature, shown in Table 2.

Baseline Methods

We re-implement the following sentiment classification algorithms as the baseline systems:

- SSWE: Tang et al. (2014), who apply SSWE as features for the model of SVM Twitter sentiment classification.
- NRC: the state-of-art system in the SemEval 2013 Twitter Sentiment Classification Track, incorporating diverse sentiment lexicons and hand-crafted features (Mohammad, Kiritchenko, and Zhu 2013).

Experimental Results

Single-prototype word embeddings Table 3 shows the macro-F scores of different embeddings. C&W is relatively weak because it does not explicitly exploit sentiment or topic information, resulting in words with opposite polarities (i.e. good and bad) being mapped to close word vectors. TEWE improves the performance compared with C&W, and the main reason is that the TEWE model contains the topic information of each tweet. Our finding is consistent with Xiang and Zhou (2014), in that topic information improves sentiment classification. NRC implements a variety of features and reaches 84.73% in macro-F score, demonstrating the importance of a better feature representation for Twitter sentiment classification. Compared with SSWE, we achieve 85.34% by using TSWE as features, which indicates the effectiveness of topic information in the word embedding space for Twitter sentiment classification.

Multi-prototypes word embeddings For multiple word embeddings, Table 3 shows the performance using different types of baseline embeddings. We can see that M-C&W gives a 4.62% improvement in micro-F score compared with C&W. In addition, M-TEWE and M-TSWE give the better performance than TEWE and TSWE, which demonstrates the effectiveness of multiple prototype word embeddings. M-TSWE gives a 86.83% macro-F score, outperforming the current best model SSWE (84.98%), which again shows the effectiveness of multi-prototype word emebeddings.

To achieve the best performance, we build a combined model in the classifier level to directly unify M-TSWE and NRC. For each model, we calculate the probability $P(c|x)$ for both class c and each tweet x in dataset. After that, we calculate the probability estimate of the union model by:

$$P_{comb}(c_i|x) = 0.5 \times P_{TMWE}(c_i|x) + 0.5 \times P_{NRC}(c_i|x) \tag{9}$$

Finally, the combined model predicts the sentiment class of a tweet x to be the class c with higher probability estimate $P_{comb}(c|x)$. Shown in Table 3, the combined model further improves the performance to 87.97%, which is the best performance on the SemEval 2013 dataset.

The influence of n-gram, topic and sentiment For M-TSWE, we tune the hyper parameters α and β on the development set. As given in Equation 6, α is the weighting score of sentiment loss of model, and β is the weighting score of topic loss of model. M-TSWE is trained from 10M distant-supervised tweets.

Figure 3 shows the macro-F scores of M-TSWE with different α and β on our development set. We can see that M-TSWE gives better performance when $\alpha = \beta = 0.3$, which balances n-gram, sentiment information and topic information. The model with $\alpha = 0$ and $\beta = 0$ stands for M-C&W. The sharp decline reflects the importance of sentiment information and topic information in learning word embeddings for Twitter sentiment classification. Compared with M-C&W model, The model with $\alpha = 0.5$ and $\beta = 0.5$ gives lower performance, which shows the importance of n-gram information. As a result, we set α and β to 0.3 in our final experiments.

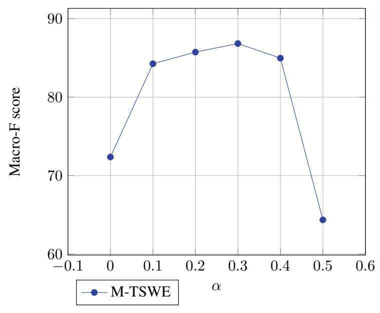

Figure 3: Macro-F scores of M-TSWE with different α and β, where 0 denotes $\alpha = \beta = 0$, and 0.1 denotes $\alpha = \beta = 0.1$, etc.

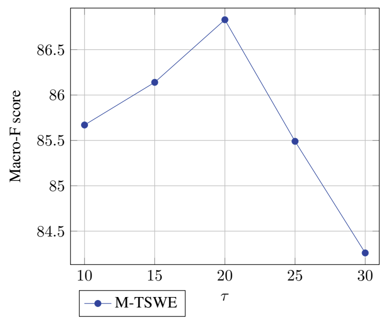

Figure 4: Macro-F scores with different topic numbers τ.

Effect of topic number on M-TSWE In this paper, we obtain the topic distribution for all tweets based on LDA. We examine the impact of the different topic numbers (from 10 to 30) on the overall performance.

Figure 4 shows the macro-F score of M-TSWE with different topic numbers τ on our development set. The performance gradually increases as the topic number increases from 10 to 20. However, the performance reduces as the topic number decreases from 20 to 30. As a result, we set the number to 20 in our final experiments.

M-TSWE merges clusters by the distance δ. We experiment with different δ (from 0.05 to 0.25) measuring the classification performance. Results show that when δ is set to 0.16, M-TSWE gives the best performance. As a result, we set 0.16 to δ in the final model. Here, we omit the detailed figures due to space limitations.

Word Similarity of Sentiment Lexicons

To show the effectiveness of topic information for Twitter sentiment classification, we measure word similarity in the embeddings for sentiment lexicons. Following Tang et al. (2014), the performance is measured by the accuracy of polarity consistency between each sentiment word and its top 100 closest words in the sentiment lexicon.

Experimental Setup Two widely-used sentiment lexicons, HL (Hu and Liu 2004) and MPQA (Wilson, Wiebe, and Hoffmann 2005), are used for evaluating the quality of word embeddings. In each lexicon, the words that do not appear in the embedding lookup table are removed. The details of the lexicons used in this paper are listed Table 4.

Lexicon	Positive	Negative	Total
HL	1,329	2,644	3,973
MPQA	1,926	2,811	7,737

Table 4: Statistics of the sentiment lexicons.

Lexicon	Random	C&W	TEWE	SSWE	TSWE
HL	50.00	64.72	75.01	77.18	**79.96**
MPQA	50.00	59.33	68.71	71.39	**72.07**

Table 5: Polarity consistency of words in different lexicons.

Results The results are shown in Table 5. The accuracy of *random* vectors is 50%. Our proposed model TSWE gives 79.96% and 72.07% in HL and MPQA, respectively, which outperform the current best model SSWE (Tang et al. 2014). Results show that topic and sentiment-enriched word embeddings can distinguish words with opposite sentiment polarity, which are not well resolved by the C&W model. TEWE also gives better performance compared with C&W. The above demonstrates the effectiveness of topic information in word embeddings for Twitter sentiment classification.

Case Study We present some cases on word embeddings generated by the experiments. We define the sentiment distance between two embeddings x_1 and x_2 as:

$$d_{sen} = 1 - cos < x_1, x_2 > \tag{10}$$

Among the word embeddings generated by TSWE, SSWE and C&W, we select three pairs of antonyms and calculated their sentiment distance, which is shown in Table 6. For *sane* and *insane*, the contexts are alike, and the global tweet sentiment information hardly distinguishes them, because the two words occur in both positive and negative environments. However, when topic information is added, the distance between the two words is enlarged because they have different usages in different topics. *insane* is often used in daily topics such as movie and sports, while *sane* is often used in formal topics like economy and health. The other two groups of antonyms are similar, illustrating that topic information can increase the sentiment distance between the words with opposite polarities.

Conclusion

We proposed to learn topic-enriched multi-prototype word embeddings for Twitter sentiment classification. First, two neural networks are developed to learn different word embedding by integrating topic and sentiment information to fully capture the topic. Second, multiple word embeddings were generated for every word. Finally, a convolutional neural network model was used to incorporate multiple word embeddings to conduct Twitter sentiment classification. Experiments on Twitter sentiment analysis on standard SemEval 2013 data showed the effectiveness of topic information and multi-prototype embeddings.

Words	C&W	SSWE	TSWE
$sane, insane$	0.1997	0.2904	**0.4443**
$waste, improve$	0.1761	0.2108	**0.2639**
$complex, simple$	0.1938	0.3318	**0.6763**

Table 6: Distance between antonyms in different models.

Acknowledgements

We thank all reviewers for their detailed comments. This work is supported by the National Natural Science Foundation of China (No. 61173062, 61373108, 61133012, 61202193), the Major Projects of the National Social Science Foundation of China (No. 11&ZD189), the Key Program of Natural Science Foundation of Hubei, China (Grant No.2012FFA088), and Singapore Ministry of Education (MOE) AcRF Tier 2 grant T2MOE201301.

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

dos Santos, C. N., and Gatti, M. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING*.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12:2121–2159.

Feldman, R. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM* 56(4):82–89.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, 249–256.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1:12.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD*, 168–177.

Hu, X.; Tang, J.; Gao, H.; and Liu, H. 2013. Unsupervised sentiment analysis with emotional signals. In *WWW*, 607–618.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*, 873–882.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent twitter sentiment classification. In *Proceedings of ACL*, 151–160.

Labutov, I., and Lipson, H. 2013. Re-embedding words. In *Proceedings of ACL*, 489–493.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1):1–167.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, 142–150.

Mohammad, S. M.; Kiritchenko, S.; and Zhu, X. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, 321–327.

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

Owoputi, O.; OConnor, B.; Dyer, C.; Gimpel, K.; and Schneider, N. 2012. Part-of-speech tagging for twitter: Word clusters and other advances. *School of Computer Science, Carnegie Mellon University, Tech. Rep.*

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of ACL*, 79–86.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, volume 1631, 1642.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin, B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of ACL*, volume 1, 1555–1565.

Tian, F.; Dai, H.; Bian, J.; Gao, B.; Zhang, R.; Chen, E.; and Liu, T.-Y. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING*, 151–160.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, 384–394.

Vanzo, A.; Croce, D.; and Basili, R. 2014. A context-based model for sentiment analysis in twitter. In *Proceedings of COLING*, 2345–2354.

Vo, D.-T., and Zhang, Y. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*, 1347–1353.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of EMNLP*, 347–354.

Xiang, B., and Zhou, L. 2014. Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In *Proceedings of ACL*, 434–439.

Zhang, M., and Zhang, Y. 2015. Combining discrete and continuous features for deterministic transition-based dependency parsing. 1316–1321.