Representation Learning for Aspect Category Detection in Online Reviews

Xinjie Zhou, **Xiaojun Wan**[*] and **Jianguo Xiao**
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China
{zhouxinjie, wanxiaojun, xiaojianguo}@pku.edu.cn

Abstract

User-generated reviews are valuable resources for decision making. Identifying the aspect categories discussed in a given review sentence (e.g., "food" and "service" in restaurant reviews) is an important task of sentiment analysis and opinion mining. Given a predefined aspect category set, most previous researches leverage hand-crafted features and a classification algorithm to accomplish the task. The crucial step to achieve better performance is feature engineering which consumes much human effort and may be unstable when the product domain changes. In this paper, we propose a representation learning approach to automatically learn useful features for aspect category detection. Specifically, a semi-supervised word embedding algorithm is first proposed to obtain continuous word representations on a large set of reviews with noisy labels. Afterwards, we propose to generate deeper and hybrid features through neural networks stacked on the word vectors. A logistic regression classifier is finally trained with the hybrid features to predict the aspect category. The experiments are carried out on a benchmark dataset released by SemEval-2014. Our approach achieves the state-of-the-art performance and outperforms the best participating team as well as a few strong baselines.

Introduction

User-generated reviews play an important role in each individual's decision making process. Opinion mining and sentiment analysis for online reviews has become a trending research topic in both academic and industrial fields since early 2000 (Turney, 2002; Pang et al., 2002). Aspect category detection is one of the opinion mining tasks which aims to identify the aspect categories discussed in a review sentence. A set of aspect categories are usually predefined which makes the task become a multi-label classification problem. For example, in SemEval-2014, {"service", "food", "price", "ambience", "anecdote /miscellaneous"} is defined as the aspect category set for restaurant reviews. In the sentence "*Service is top notch.*", "service" should be detected as the aspect category. Opinion without knowing the target is of limited use (Liu, 2011). Identifying the aspect category helps to get target-dependent sentiment and contributes to aspect-specific opinion summarization.

Previous researches have proposed several models to address this task and SVM classification is one of the most popular ones (Ganu et al.,2009; Kiritchenko et al 2014). These existing methods have shown the significance of lexical information in aspect category detection. However, the unigram or n-gram based features usually use one-hot representations and fail to capture semantic relations between different words. Words that appear in the training data cannot provide any information if it does not appear in the test data. Associations between different words cannot be quantitatively measured via the one-hot vectors. To overcome the shortcomings of the existing studies, we propose a representation learning approach for aspect category detection.

Firstly, we propose a semi-supervised word embedding algorithm. It captures semantic relations between words, relations between words and aspects, and relations between sentiment words and aspects. After obtaining the word vectors, we average all the word vectors in a sentence as its continuous representation (Huang et al., 2012). Different from existing works that directly learn supervised classifiers based on the sentence vectors (Tang et al., 2014b), we propose to generate deeper and hybrid features which help to boost the performance. Two different kinds of neural networks are used for learning shared features and aspect-specific features respectively. We get the hybrid features by concatenating them together. The logistic regression classifier trained on the hybrid features achieves the state-of-the-art performance on the benchmark dataset

[*] Xiaojun Wan is the corresponding author.

released by SemEval-2014[1]. The performance is higher than that of the best participating team as well as a few strong baselines.

The main contributions of the study are summarized as follows, 1) we propose a representation learning approach for aspect category detection which achieves the state-of-the-art performance on a benchmark dataset. 2) We propose a semi-supervised word embedding algorithm which captures semantic relations between words, relations between words and aspects, and relations between sentiment words and aspects in a unified framework. 3) We generate deeper and hybrid features by using two different kinds of neural networks. It shows better performance than either the shared features or the aspect-specific features.

Problem Definition

Aspect category detection is a major evaluation task of SemEval-2014 (Semantic Evaluation), which attracted 18 teams global wide to participate in. Aspect category detection aims to identify the aspect categories discussed in a given review sentence. Formally, an aspect category set A = {a_1, a_2, a_3 … a_N} which contains N categories is predefined for a product domain. For a review dataset D = {s_1, s_2, s_3 … s_K} which contains K sentences, we need to predict a binary label vector $y_i \in R^{N\times 1}$ $(1 \leq i \leq K)$ for each sentence. Each value in y_i indicates whether the sentence s_i is discussing an aspect category or not. Specifically, $y_i^m = 1$ $(1 \leq i \leq K, 1 \leq m \leq N)$ means that sentence s_i contains the aspect category a_m and $y_i^m = 0$ otherwise. In SemEval-2014, the restaurant review dataset is used for evaluation. Five aspect categories are predefined for the domain, i.e. A = {“food”, “service”, “price”, “ambience”, “anecdote /miscellaneous” (“a/m” for short)}.

Aspect category is sometimes the hypernym of the aspects in a sentence. For example, in the review “the steak was mouthwatering!”, the aspect category “food” is the hypernym of the aspect “steak”. Meanwhile, some review sentences do not contain any aspect but are still expressing an opinion towards a target, such as the following one “Everything is tasty and well-portioned.” Aspect category detection can solve such implicit aspect extraction problem (Hai, Chang, and Kim, 2011) elegantly by regarding it as a classification task. The implicit aspect can be detected if it is included in the aspect category set.

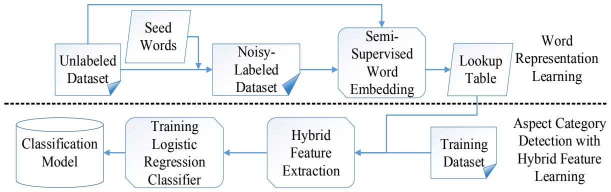

Seed Words
Unlabeled Dataset
Noisy-Labeled Dataset
Semi-Supervised Word Embedding
Lookup Table
Word Representation Learning
Classification Model
Training Logistic Regression Classifier
Hybrid Feature Extraction
Training Dataset
Aspect Category Detection with Hybrid Feature Learning

Figure 1. The framework of our proposed approach

Our Proposed Approach

In this section, we describe our representation learning approach for aspect category detection. We first propose a semi-supervised word embedding algorithm to obtain word vectors. Secondly, we acquire deeper and hybrid features through neural networks for supervised prediction. The general framework of our approach is shown in Figure 1.

Word Representation Learning

As shown in Figure 1, the word representation learning method leverages a large unlabeled review dataset. We use several seed words to assign category labels. The noisy-labeled data help our algorithm to obtain aspect-specific word embedding. The rest of the dataset remains as unlabeled. It helps to capture semantic relations between different words. We also extract sentiment word and aspect word pairs though dependency patterns. They enable our algorithm to learn relations between sentiment words and aspects.

To capture the semantic associations, we follow the strategy of word2vec. The skip-gram model of word2vec tries to maximize the probability of predicting a context word from a center word (Mikolov et al, 2013),

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\mathrm{T}} v_{w_I})}{\sum_{w=1}^{W} \exp({v'_{w}}^{\mathrm{T}} v_{w_I})}$$

where w_O and w_I are the input (word) and output (context word); v_* and v'_* denotes the vectors associated with a word and a context word, respectively. The formulation forces words with similar contexts to get similar vector representations. However, it is impractical because the partition function grows linearly with the vocabulary size which could be hundreds of thousands.

A computationally efficient approximation for the above equation is negative sampling which has been used in Gutmann and Hyvarinen (2012) and Mnih and Teh (2012). A log-bilinear model is used to predict whether two words are in the same context and the loss function becomes,

$$L_1 = -\log\sigma\left({v'_{w_O}}^{\mathrm{T}} v_{w_I}\right) - \sum_{i=1}^{n} E_{w_i \sim P_n(w)}\left[\log\sigma(-{v'_{w_i}}^{\mathrm{T}} v_{w_I})\right]$$

where σ(·) is the sigmoid function, w_O is the actual context word and w_i is the negative sample from the noise distribution $P_n(w)$. Mikolov et al. (2013) set $P_n(w)$ as the 3/4rd power of the unigram distribution which outperforms the unigram and the uniform distribution significantly.

Since the skip-gram model only captures word semantic relations, we hope that words associated with different aspects fall into different positions in the vector space. A straight-forward strategy is adding supervision to the

[1] http://alt.qcri.org/semeval2014/task4/

learning algorithm. While manual annotation can be time-consuming and expensive, we use distant supervision instead. Several seed words are selected to assign category labels to sentences which contain these seed words. For minimizing the human interaction, we only use the category names as the seed words to get noisy labels. For example, all the review sentences which contain "food" are regarded to fall into the "food" category. We do not collect any labeled data for "a/m" because this category is quite ambiguous and may involve any aspect aside from the other four categories. The labeled sentences are further sampled to balance the amounts of sentences in different categories.

For each sentence s_i with a noisy label vector y_i, we use logistic regression to classify the aspect category and adopt the cross entropy loss function,

$$L_2 = \sum_{1 \le m \le N} -y_i^m \log \sigma\left({x_m}^{\mathrm{T}} V_i + b_m\right) - \left(1 - y_i^m\right) \log \sigma\left(-{x_m}^{\mathrm{T}} V_i - b_m\right)$$

where x_m is the weight vector for the features, b_m is the bias and V_i is the sentence vector for s_{i}. We simply average all the word vectors v_w $(w \in s_i)$ to get the sentence vector.

Apart from the word-aspect association, capturing the relation between sentiment words and aspects can also be useful for aspect category detection (McAuley et al., 2012). Many sentiment words such as "delicious" and "tasty" are aspect-specific because they can only describe "food". For each sentiment word – aspect word pair, we hope that the embedding algorithm can make them fall into the same side of a classification hyperplane defined by x_m and b_m (the same classification hyperplane for the sentences). We use dependency patterns to extract word pairs such as "food" and "delicious" from the phrase "food is delicious". The pattern is defined as "Noun → SBJ → W ← PRD ← Adjective" where "Noun" and "Adjective" represent the part-of-speech tags, "SBJ" and "PRB" are dependency relations, "W" can be any word. The noun word and the adjective word together are treated as a word pair. For each word pair (w_1, w_2), their probabilities of falling into an aspect category should be similar which brings the following loss function,

$$L_3 = \left[\sigma\left({x_m}^{\mathrm{T}} v_{w_2} + b_m\right) - \sigma\left({x_m}^{\mathrm{T}} v_{w_1} + b_m\right)\right]^2$$

The above equation tends to narrow the distance between $\sigma\left({x_m}^{\mathrm{T}} v_{w_2} + b_m\right)$ and $\sigma\left({x_m}^{\mathrm{T}} v_{w_1} + b_m\right)$.

Finally, we learn the word vectors by aggregating all the above loss functions,

$$minimize\ L = L_1 + L_2 + L_3$$

The parameters are learned through stochastic gradient descent. We omit the details due to limited space.

Aspect Category Detection with Hybrid Feature Learning

We simply average all the word vectors to obtain the sentence vector. Different from most of the previous researches that directly learn supervised classifiers on the sentence vector, we choose to get deeper and hybrid features through different two-layer feed-forward neural networks.

The input of a neural network is a sentence vector V_{i} which represents the sentence s_{i} in the training dataset. We adopt sigmoid as the activation function and get the hidden units H_i as

$$H_i = \sigma(W_1 V_i + B_1)$$

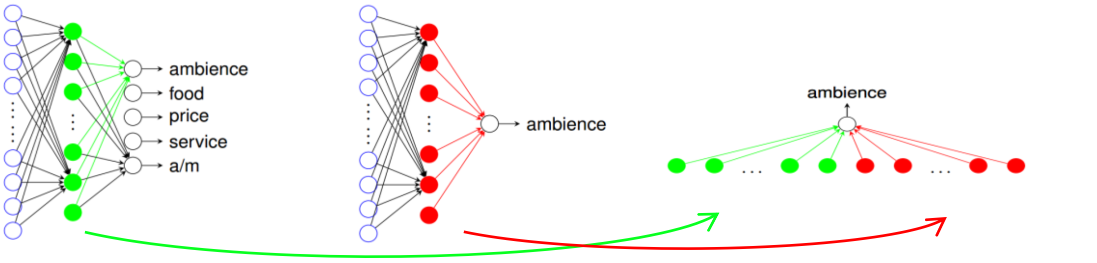
and the output layer O_i as

$$O_i = \sigma(W_2 H_i + B_2)$$

where W_1 and B_1 are the parameters of the first layer, W_2 and B_2 are the parameters of the second layer.

We introduce two different settings of the neural network to obtain the shared and aspect-specific features, respectively. In the first setting, a two-layer neural network is trained to fit all the aspect categories simultaneously as shown in Figure 2.a. The neural network outputs five binary variables to represent five aspect categories. The output O_i represents the label vector y_i for the sentence s_i. We use back-propagation to learn the parameters. In this setting, the model learns the same features (i.e. the hidden layer) to predict all the aspect categories. Therefore, we describe the hidden layer as shared features.

In the second setting, we use five different two-layer neural networks to predict each of the five aspect



(a) Learning shared features *(b) Learning aspect-specific features* *(c) Hybrid features and weight initialization*

Figure 2. Learning Deeper and Hybrid Features

categories. In this setting, the output O_i corresponds to one of the five values in y_i. The hidden layer for each aspect is different from each other because they are trained separately (Figure 2.b). The features only need to adapt to one of the aspect category. Therefore, we describe the hidden layer as aspect-specific features.

Finally, we concatenate the shared and aspect-specific features to form our hybrid features. A 2-class logistic regression classifier is trained on the hybrid features for each of the aspect (Figure 2.c). We use the weights learned from the neural networks to initialize the weights here. Afterwards, the weights are fine-tuned through stochastic gradient descent. The weight initialization step helps the training procedure converge much faster.

Experiments

Dataset

We used the restaurant review dataset released by SemEval-2014 which modified and extended the dataset of Ganu et al. (2009). The training dataset contains 3,041 sentences and the test dataset contains 800 sentences. We show the number of sentences in each category in Table 1.

Category	# of sentences	
	Training	Test
food	1232	418
price	321	83
service	597	172
ambience	431	118
a/m	1132	234

Table 1. Statistics of the SemEval-2014 Restaurant Review Dataset

Additionally, we collected an Extended Restaurant Review Dataset to learn the word representations. Part of the dataset is provided by Yelp Dataset Challenge[2]. The rest of the dataset is crawled from Citysearch[3]. We use the category names "food", "price", "service" and "ambience/ambiance" as seed words to obtain lots of noisy-labeled sentences (Purver and Battersby, 2012). Mate-tools (Bohnet, 2010) is used to parse the dataset and get adjective-noun word pairs via the dependency pattern. Note that the Extended Restaurant Review Dataset is only used for learning better word representations. The final classification model is trained on the SemEval-2014 Restaurant Review Dataset. The detailed statistics of the dataset are shown in Table 2.

# of unlabeled sentences	# of noisy-labeled sentences	# of word pairs
8,324,813	1,214,762	1,790,421

Table 2. Statistics of the Extended Restaurant Review Dataset

Experiment Setup

In word representation learning, we set the vector size as 500 and the context windows as 5. The learning rate is set to 0.025 following word2vec and it declines with the training procedure. All the word vectors (v_i) and the classification weights (x_m and b_m) are initialized randomly between -0.5 and 0.5.

For learning the hybrid features, we use two 2-layer neural networks, both of which contain 50 hidden units. We use the labeled dataset (not the noisy-labeled dataset) to train them through back-propagation. The mini-batch stochastic gradient descent is used to update the parameters. The batch size is set to 50. The training procedure is run for 500 epochs when the training error becomes steady.

Results and Analysis

Baseline Methods

Firstly, we compare our method with several traditional classification algorithms.

KNN: This is the baseline provided by SemEval-2014 (Pontiki et al., 2014). For each test sentence s_i, k most similar training sentences are first found. The Dice coefficient is used to measure the sentence similarity. Then, s_i is assigned the m most frequent aspect category labels of the k retrieved sentences; m is the most frequent number of aspect category labels per sentence among the k sentences.

NB, LR and **SVM**: We use Naïve Bayes, Logistic Regression and Support Vector Machine as the classification algorithms with unigram and bigram features.

SVM-DS: We incorporate distant supervision into the SVM model. Both manually labeled data and the noisy-labeled data are used for training.

NRC: This is the best system in the evaluation (Kiritchenko et al., 2014). They also adopt SVM as the classification algorithm. The features include n-grams, stemmed n-grams, character n-grams, non-contiguous n-grams, word cluster n-grams and lexicon features. **NRC-Lexicon** is the result without the lexicon features.

SemEval-Avg: The average result of all the systems in SemEval-2014.

We also compare our method with existing word embedding algorithms including **C&W** (Collobert and Weston, 2008), **word2vec** (Mikolov et al, 2013), **HLBL** (Mnih and Hinton, 2008) and **GloVe** (Pennington et al. 2014). Pre-trained word vectors are publicly available on

[2] http://www.yelp.com/dataset_challenge
[3] http://www.citysearch.com/

the web for all these algorithms.[4] Beside, word2vec and GloVe provide training code so that we can re-run the model on our Extended Restaurant Review Dataset (**word2vec-re** and **GloVe-re**). After obtaining the word vectors, we extract hybrid features to train the classifiers.

Method	F1-Score
KNN	63.89
LR†	66.01
NB†	66.70
SVM†	**80.81**
SVM-DS†	70.97
SemEval-Avg	73.79
NRC-Lexicon	84.08
NRC (Best SemEval System)	**88.57**
HLBL†	69.69
C&W†	72.55
GloVe†	81.12
GloVe-re*†	84.55
word2vec†	83.31
word2vec-re*†	**87.67**
Ours	**90.10**

*Table 3. Performance on the benchmark dataset. * denotes that the word embedding method is trained on our restaurant review dataset. † denotes significant statistical difference between the method and our approach (p≪0.01 in Sign-test). SemEval do not release the detailed results, so the significance test is not carried out for KNN, NRC and NRC-Lexicon.*

Results

We use micro F1-score of all the category labels as the evaluation metric. Table 3 shows the comparison results of all the baseline methods and our approach. LR, NB and SVM are the most widely used classification algorithms. Among these three methods, SVM outperforms LR and NB by a large margin. It achieves the F1-score over 80 by using unigram and bigram features. However, when the noisy-labeled data are used for training, the performance declines. It shows that the noisy-labeled data cannot improve the result when they are directly used to train the classifier. The best system in SemEval-2014 is NRC which also relies on SVM. Besides the textual features, they use an additional lexicon which contains the associations between words and aspects. The lexicon helps to boost the performance from 84.08 (NRC-Lexicon) to 88.57. Even without the lexicon, the performance is still higher than our SVM baseline. That is because more complicated features such as word cluster n-gram are used in their method. It also shows that feature engineering is a crucial step for improving the performance.

Four different word representation learning algorithms are used for comparison here. After obtaining the word vectors, we use the two-layer neural networks to obtain hybrid features and train a logistic regression classifier for each aspect as describe in Section 4. From the experimental results, we can find that GloVe and word2vec outperform the other two word embedding algorithms by a large margin on our task. Our collected restaurant review dataset helps to improve the performance of both word2vec and GloVe remarkably. The hybrid features extracted from word2vec-re achieve comparable results with NRC. Compared to word2vec-re, our model incorporates distant supervision and captures associations between sentiment words and aspects. It achieves the state-of-the-art performance on the dataset with the F1-score of 90.10. Overall, our representation learning approach outperforms the traditional hand-crafted features as well as existing word embedding algorithms.

Hybrid Feature Analysis

In this study, we propose to extract deeper and hybrid features after obtaining the word vectors. The two-layer neural networks compress the original features (500 to 50) and improve the performance remarkably.

Method	F1-Score
Averaged Word Vectors	85.36
Aspect-Specific Features	89.51
Shared Features	89.30
Hybrid Features	90.10

Table 4. Feature analysis results.

Tables 4 shows the comparison results of different features. Logistic regression is used to train the classifiers for all of them. When the classifier is directly trained on the averaged vector of all words in a sentence, the F1-score is 85.36. It has already outperformed the SVM method with textual features. The two-layer neural networks provide an increase in F1-score of 4 points for both aspect-specific and shared features. Although the dimension of the compressed features are only 1/10 of the original ones, the performance is largely improved. It shows that our method effectively compresses the features and improves the generalization ability. The aspect-specific features and shared features get similar F1-score. By combining the two different kinds of features together, our hybrid features get the highest performance. In the experiment, we also find that the improvement is very stable during the training procedure. In all the training iterations of the neural networks, the hybrid features always outperform the aspect-specific features and shared features by 0.5~2 points of F1-score.

Sensitivity of the Vector Size

In this subsection, we analyze the effect of the vector size in wording embedding algorithms. We choose different vector sizes ranging from 50 to 1000. The performances of

[4] Word2vec: http://code.google.com/p/word2vec/. GloVe: http://nlp.stanford.edu/projects/glove/. C&W and HLBL: http://metaoptimize.com/projects/wordreprs/ (Turian et al., 2010).

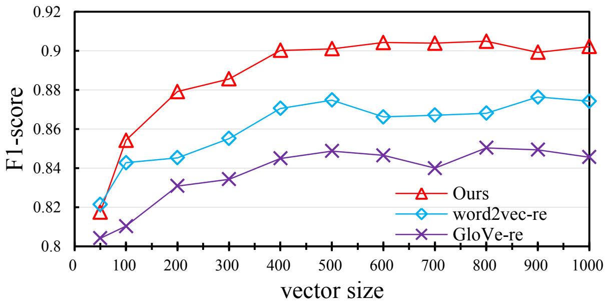


Figure 3. Effect of word vector size

word2vec-re, GloVe-re and our method on aspect category detection are plotted in Figure 3.

The three curves show similar patterns. We can see that the F1-score improves rapidly with the increase of the vector size when it is less than 400. When the vector size grows larger, the performance becomes steady. Since the training time of the word embedding algorithm is linear with the vector size, choosing it between 400 and 600 helps to balance the algorithm performance and the time complexity. We can also see that our approach always achieves higher F1-score than either word2vec-re or GloVe-re when the vector size is above 50.

Related Work

Aspect-based sentiment analysis

Aspect-based sentiment analysis is a fine-grained opinion mining task. In product reviews, the opinion target can be decomposed into entity and its aspects. Aspect-based sentiment analysis aims to find the aspects and the corresponding sentiment toward them (Qiu et al., 2011; Liu et al., 2014). It requires deeper NLP capabilities and produces a richer set of results.

Aspect extraction has been widely studied since the pioneering work of Hu and Liu (2004). In recent years, topic modeling has become the mainstreaming approach to deal with the problem. These methods simultaneously extract aspects and categorize them into several topics. Titov et al (2008) proposed the multi-grain topic model. The model uses the global topic to capture aspect-independent words and uses local topics to capture aspect-specific words. Zhao et al. (2010) and Mukherjee et al. (2012) extended the multi-grain topic model and separated aspect words and the corresponding sentiment words into different topics.

Aspect category detection is a special case of aspect-based sentiment analysis. Instead of extracting aspects, an aspect category set is given in advance and the goal is to classify each review sentence into one or more aspect categories. Ganu et al. (2009) directly used SVM to train one vs. all classifiers on restaurant reviews. They only used stem word as features because sophisticated features did not bring remarkable improvement. Kiritchenko et al. (2014) applied the same algorithm but exploited a Yelp word-aspect association lexicon to boost the performance. Their system achieved the top ranking in the aspect category detection subtask of SemEval-2014. McAuley (2012) proposed a discriminative model to predict product aspect. They used two kinds of parameters to encode the word association. One of them learns which words are associated with each of the aspects. The other learns which words are associated with each star rating.

Learning continuous word representation

Learning vector space representations for natural language texts has succeeded in capturing fine-grained semantic and syntactic relations. Bengio et al. (2003) proposed a neural network language model which learned simultaneously a distributed representation for each word along with the probability function for word sequences. Afterwards, word embedding has become a hot research topic to represent semantics in a distributed manner (Mnih and Hinton, 2008; Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014).

Besides the unsupervised word embedding algorithms, learning task-specific word embedding has shown promising performance on many tasks. Labutov and Lipson (2013) proposed to fine-tune the existing word vectors based on a labeled dataset. Tang et al. (2014a) presented the sentiment-specific word embedding method which can separate words like “good” and “bad” to opposite ends. Tang et al. (2014b) learned a word embedding model which helped to classify a word into positive, negative or neural.

Conclusion and Future Work

In this study, we propose a representation learning approach for aspect category detection. We show that the semi-supervised word embedding algorithm along with the hybrid feature extraction approach brings state-of-the-art performance for aspect category detection. For future work, we will test our approach on more datasets and in different languages. The framework can also be extended to predict both aspect categories and aspect ratings.

Acknowledgments

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China (2014AA015102, 2012AA011101), National Natural Science Foundation of China (61170166, 61331011) and Beijing Nova Program (2008B03).

References

Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., and Gauvain, J. L. 2006. Neural probabilistic language models. In *Innovations in Machine Learning* (pp. 137-186). Springer Berlin Heidelberg.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

Ganu, G., Elhadad, N., and Marian, A. 2009. Beyond the Stars: Improving Rating Predictions using Review Text Content. In *WebDB*.

Gutmann, M. U., and Hyvärinen, A. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The Journal of Machine Learning Research*, 13(1), 307-361.

Hai, Z., Chang, K., and Kim, J. J. 2011. Implicit feature identification via co-occurrence association rule mining. In *Computational Linguistics and Intelligent Text Processing* (pp. 393-404). Springer Berlin Heidelberg.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168-177). ACM.

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 the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 873-882). Association for Computational Linguistics.

Labutov, I., and Lipson, H. 2013. Re-embedding words. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics* (pp. 489-493).

Liu Kang, Xu Liheng, and Zhao Jun. Extracting Opinion Targets and Opinion Words from Online Reviews with Graph Co-ranking. In *Proceedings of ACL 2014*, Baltimore, USA, June 22-27.

Kiritchenko, S., Zhu, X., Cherry, C., and Mohammad, S. M. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval* 2014, 437.

Liu, B. 2011. Opinion mining and sentiment analysis. In *Web Data Mining* (pp. 459-526). Springer Berlin Heidelberg.Qiu G. 2009. Double Propagation

McAuley, J., Leskovec, J., and Jurafsky, D. 2012. Learning attitudes and attributes from multi-aspect reviews. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on* (pp. 1020-1025). IEEE.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111-3119).

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems* (pp. 1081-1088).

Mnih, A., and Teh, Y. W. 2012. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint* arXiv:1206.6426.

Mukherjee, A., and Liu, B. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 339-348). Association for Computational Linguistics.

Pang, B., Lee, L., & Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86). Association for Computational Linguistics.

Pennington J., Socher R., and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP).

Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, Dublin, Ireland.

Purver, M., & Battersby, S. 2012. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 482-491). Association for Computational Linguistics.

Qiu, G., Liu, B., Bu, J., and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1), 9-27.

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems* (pp. 801-809).

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. 2014a. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 1555-1565).

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. 2014b. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics* (pp. 172-182)

Titov, I., and McDonald, R. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web* (pp. 111-120). ACM.

Turian, J., Ratinov, L., and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 384-394). Association for Computational Linguistics.

Turney, P. D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.

Zhao, W. X., Jiang, J., Yan, H., and Li, X. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (pp. 56-65). Association for Computational Linguistics.