NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features

Zhiqiang Toh
Institute for Infocomm Research
1 Fusionopolis Way
Singapore 138632
ztoh@i2r.a-star.edu.sg

Jian Su
Institute for Infocomm Research
1 Fusionopolis Way
Singapore 138632
sujian@i2r.a-star.edu.sg

Abstract

This paper describes our system submitted to Aspect Based Sentiment Analysis Task 5 of SemEval-2016. Our system consists of two components: binary classifiers trained using single layer feedforward network for aspect category classification (Slot 1), and sequential labeling classifiers for opinion target extraction (Slot 2). Besides extracting a variety of lexicon features, syntactic features, and cluster features, we explore the use of deep learning systems to provide additional neural network features. Our system achieves the best performances on the English datasets, ranking 1st for four evaluations (Slot 1 for both restaurant and laptop domains, Slot 2, and Slot 1 & 2).

1 Introduction

Sentiment analysis and opinion mining have gained increasing interests in recent years due to the continuous growing of user-generated content on the Internet. Traditionally, the primary focus of the research has been on the detection of the overall sentiment of a sentence or paragraph. However, such approach is unable to handle conflicting sentiment for different aspects of the same entity. Hence, a more fine-grained approach, known as Aspect-Based Sentiment Analysis (ABSA), is proposed. The goal is to correctly identify the aspects of entities and the polarity expressed for each aspect.

The SemEval-2016 Aspect Based Sentiment Analysis (SE-ABSA16) task is a continuation of the same task in 2015 (Pontiki et al., 2015). Besides sentence-level ABSA (Subtask 1), it provides datasets to allow participants to work on text-level ABSA (Subtask 2). In addition, additional datasets in languages other than English are available (Pontiki et al., 2016).

We participate in Subtask 1 of SE-ABSA16, where we submitted results for Slot 1 (aspect category classification), Slot 2 (opinion target extraction), and Slot 1 & 2 (assessing whether a system correctly identifies both Slot 1 and Slot 2) for the English datasets.

Our work is based on our previous machine learning system described in Toh and Su (2015), enhanced using additional features learned from neural networks. For Slot 1, we treat the problem as a multi-class classification problem where aspect categories are predicted via a set of binary classifiers. The one-vs-all strategy is used to train a binary classifier for each category found in the training data. Each classifier is trained using a single layer feedforward network. We enhance the system by adding neural network features learned from a Deep Convolutional Neural Network system. For Slot 2, we treat the problem as a sequential labeling task, where sequential labeling classifiers are trained using Conditional Random Fields (CRF). The output of a Recurrent Neural Network system is used as additional features. To generate Slot 1 & 2 predictions, the predictions of Slot 1 and Slot 2 are combined.

The remainder of this paper is organized as follows. In Section 2, the features used in our system are described. Section 3 presents the detailed machine learning approaches. Section 4 and Section 5 show the official evaluation results and feature ablation results respectively. Finally, Section 6 summa-

Proceedings of SemEval-2016, pages 282–288,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

rizes our work.

2 Features

Our system used a variety of features which are briefly described in the following subsections. Most of the features used are the same as the features used in Toh and Su (2015).

2.1 Word

Each word in a sentence is used as a feature. Additional word context is used for different slots: for Slot 1, all word bigram context found in a sentence are also used; for Slot 2, the previous word and next word context are also used.

2.2 Name List

Two name lists of opinion targets are generated from the training data of the restaurant domain. One list contains opinion targets that frequently occur in the training data. The other list contains words that often occur as part of an opinion target in the training data.

2.3 Head Word

For each word, the head word is extracted from the sentence parse tree and is used as a feature.

2.4 Word Embeddings

Word embeddings have shown previously to be beneficial to opinion target extraction, requiring only minimal feature engineering effort (Liu et al., 2015). We trained word embeddings from two unlabeled datasets: the Multi-Domain Sentiment Dataset containing product reviews from Amazon (Blitzer et al., 2007)[1], and the user reviews found in the Yelp Phoenix Academic Dataset[2]. Additional word embeddings are also generated from the concatenation of the above two datasets.

Two different approaches are used to train the word embeddings. The first approach uses the gensim[3] implementation of the word2vec tool (Mikolov et al., 2013)[4]. We experiment with different vector sizes, window sizes, minimum occurrences and subsampling thresholds.

The second approach uses the GloVe tool (Pennington et al., 2014)[5]. By varying the minimum count, window size and vector size, different embedding files are generated. The best embedding files to use are selected using 5-fold cross validation.

2.5 Word Cluster

We further processed the embedding files described in Section 2.4 by generating K-means clusters from them. Specifically, the K-means clusters are generated using the K-means implementation of Apache Spark MLlib[6]. Different cluster sizes are tried out and the best cluster files are selected using 5-fold cross validation.

2.6 Double Propagation Name List

Besides using the training data to generate name lists, we used the unsupervised Double Propagation (DP) algorithm (Qiu et al., 2011) to generate candidate opinion targets and collect them into a list. We adjust the logical rules stated in Liu et al. (2013) to derive our own propagation rules written in Prolog. The SWI-Prolog[7] is used as the solver. One issue with our rules is that it can only identify single-word targets. Thus, we check each identified target and include any consecutive noun words right before the target.

3 Approaches

This section describes our approaches used to generate the predictions for the different slots. The machine learning system is based on our previous work (Toh and Su, 2015) and is extended to use additional neural network features.

3.1 Aspect Category Classification (Slot 1)

For each category found in the training data, a binary classifier is trained using the Vowpal Wabbit tool[8], which provides the implementation of the single layer feedforward network algorithm that we use.

Besides using the features reported previously, we enhance our existing system by using additional features from a deep learning system described below.

[1] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

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

[3] https://radimrehurek.com/gensim/

[4] https://code.google.com/archive/p/word2vec/

[5] http://nlp.stanford.edu/projects/glove/

[6] http://spark.apache.org/mllib/

[7] http://www.swi-prolog.org/

[8] https://github.com/JohnLangford/vowpal_wabbit/wiki

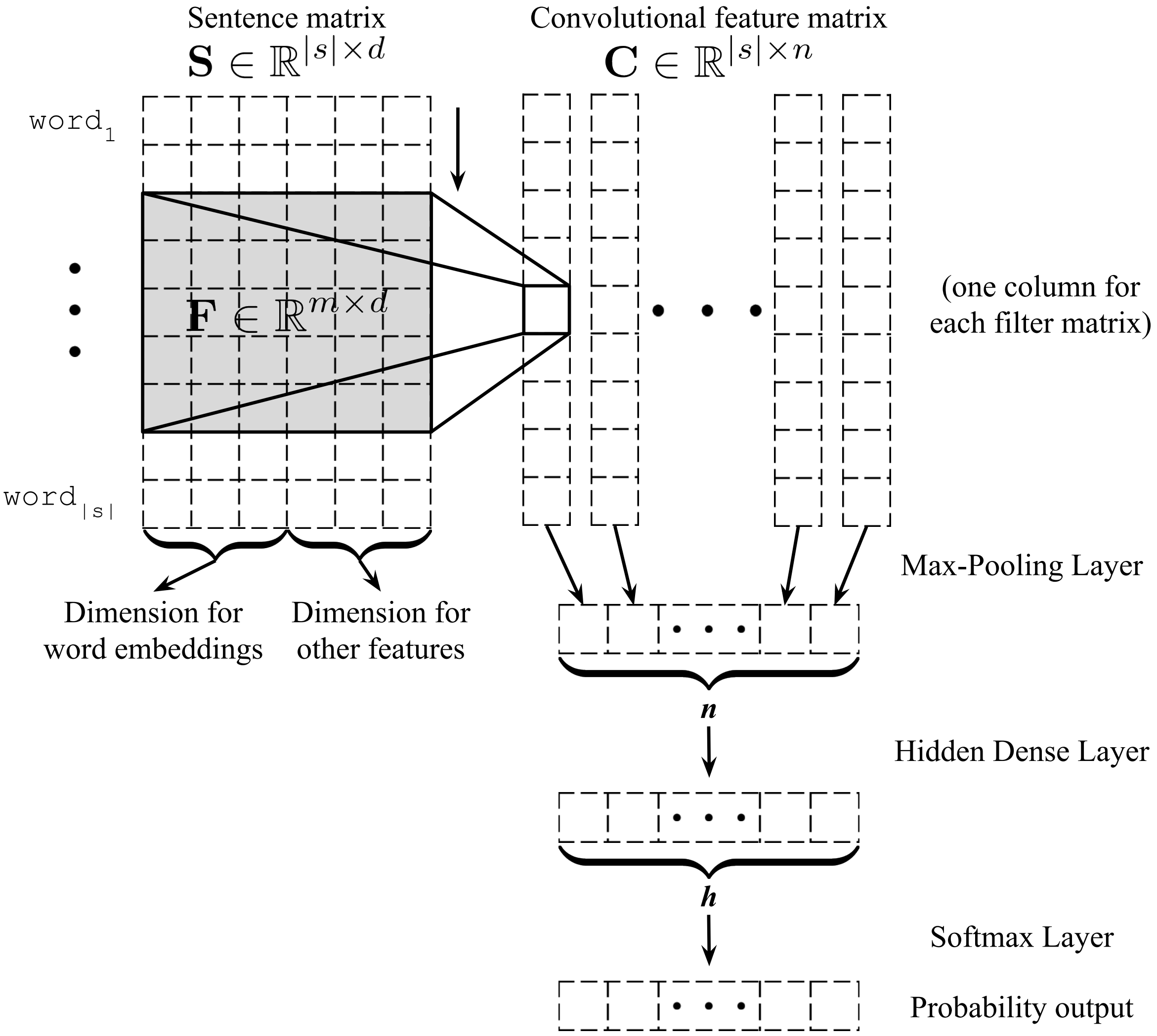
Sentence matrix
$\mathbf{S} \in \mathbb{R}^{|s| \times d}$
Convolutional feature matrix
$\mathbf{C} \in \mathbb{R}^{|s| \times n}$
word$_1$
$\mathbf{F} \in \mathbb{R}^{m \times d}$
word$_{|s|}$
(one column for each filter matrix)
Max-Pooling Layer
Dimension for word embeddings
Dimension for other features
n
Hidden Dense Layer
h
Softmax Layer
Probability output

Figure 1: The architecture of our Convolutional Neural Network.

The deep learning system is based on the Deep Convolutional Neural Network (CNN) architecture described in Severyn and Moschitti (2015). The architecture we use is shown in Figure 1.

A sentence matrix $\mathbf{S} \in \mathbb{R}^{|s| \times d}$ is built for each input sentence $\mathbf{s}$, where each row i is a vector representation of the word i in the sentence. The sentence length $|s|$ is fixed to the maximum sentence length of the dataset so that all sentence matrices have the same dimensions. (Shorter sentences are padded with row vectors of 0s accordingly.) Each row vector of the sentence matrix is made up of columns corresponding to different input features (e.g. word embedding feature, name list feature, etc.) concatenated together [9].

The input sentence matrix $\mathbf{S}$ is then passed through a series of network layer transformations, described in the following subsections.

3.1.1 Convolutional Layer

We apply a convolution operation between the input sentence matrix $\mathbf{S}$ and a filter matrix $\mathbf{F} \in \mathbb{R}^{m \times d}$ of context window size m, resulting in a column vector $\mathbf{c} \in \mathbb{R}^{|s|}$. The filter matrix $\mathbf{F}$ will slide (with a stride of 1) along the row dimension of $\mathbf{S}$, generating a value for each word in the sentence. Instead of a single filter matrix, n filter matrices are applied to

[9] Categorical features are converted to one-hot encodings.

the sentence matrix $\mathbf{S}$, resulting in a convolutional feature matrix $\mathbf{C} \in \mathbb{R}^{|s| \times n}$.

To learn non-linear decision boundaries, each element of $\mathbf{C}$ passes through the hyperbolic tangent *tanh* activation function.

3.1.2 Max-Pooling Layer

The output matrix $\mathbf{C}$ is then passed to the max-pooling layer. This layer will return the maximum value of each column.

3.1.3 Hidden Dense Layer

A hidden dense layer with h hidden units is applied to the output of the pooling layer, using Rectified Linear Unit (ReLU) as the activation function.

3.1.4 Softmax Layer

A softmax layer receives the output of the previous dense layer and computes probability distribution over the possible categories. We include an additional category "NIL" for the case where the sentence contains no aspect category. Since a sentence may contain more than one category, we output the categories whose output probability value is greater than a threshold t.

3.1.5 Network Training and Regularization

The stochastic gradient descent (SGD) algorithm is used to train the CNN network, using the back-propagation algorithm to compute the gradients. We run SGD for e epochs, where a batch size of b sentences is used. The categorical cross-entropy is used as the loss function. To prevent overfitting, the loss function is augmented with a L2 regularization term (l_2) for the parameters of the network. The Adadelta update function (with a specific decay rate ρ and constant ϵ) is used to control the learning rate.

The specific values used for the hyperparameters of the network are tuned using 5-fold cross-validation. The context window size m is set to 5. The number of filter matrices n is set to 300. The probability threshold t is set to 0.2. The number of hidden units h is set to 100. The number of epochs e is set to 50 and 100 for the restaurant and laptop domain respectively. The L2 regularization term l_2 is set to 0.01. The Adadelta decay rate ρ and constant ϵ is set to 0.95 and $1e^{-6}$ respectively.

Restaurant	
Feature	F1
Word†	0.6432
+ Head Word	0.6558
+ Name List†	0.6670
+ Word Cluster	0.7128
+ CNN Probabilities	0.7510
CNN System	0.7333

Laptop	
Feature	F1
Word†	0.5178
+ Head Word†	0.5358
+ Word Cluster	0.5463
+ CNN Probabilities	0.5983
CNN System	0.5693

Table 1: Experimental results of 5-fold cross-validation for Slot 1. Besides using the feature stated in the current row, features stated in all previous rows are also used. † indicates features used in constrained systems.

3.2 Slot 1 Features

Besides the features described in Section 2, the probability output of the CNN system is used as additional features to our multi-class classification system. The CNN system is trained on the following input features: Word Embeddings, Name List (only for the restaurant domain) and Word Cluster.

We performed 5-fold cross-validation experiments to obtain performances of the system after adding each feature group. Table 1 shows the experimental results.

We also include the 5-fold cross-validation performances if we only use the CNN system output for evaluation (last row). For both domains, the CNN system achieves better performances than the multi-class classification system without the neural network features.

However, the best performances are achieved when we used the CNN probability output as additional features to the multi-class classification system. This suggests our approach of combining two different machine learning systems is a feasible approach for the task.

3.3 Opinion Target Extraction (Slot 2)

We treat opinion target extraction as a sequential labeling task. The sequential labeling classifiers are trained using Conditional Random Fields (CRF). Such approach is similar to previous work that achieves state-of-the-art performances (Toh and Su, 2015). The implementation of CRF is provided by the CRFsuite tool (Okazaki, 2007).

Similar to our previous work, for different evaluations involving Slot 2, we train different models. For Slot 1 & 2 evaluation (multi setting), the explicit opinion targets may be classified under more than one category. Thus, a separate CRF model is trained for each category *C* found in the training data, where each model is trained using the corresponding BIO labels: "B-*C*", "I-*C*" and "O" (corresponding to start of an opinion target, continuation of an opinion target and outside respectively).

For Slot 2 evaluation (single setting), only the target span is required. Thus, all categories are collapsed into a single category (e.g. "TARGET"). A single CRF model is trained using the labels "B-TARGET", "I-TARGET" and "O".

We also enhance our existing CRF system by using the output of a Recurrent Neural Network (RNN) system as additional features.

Specifically, we implement the Bidirectional Elman-type RNN model described in Liu et al. (2015)[10]. Such a model allows long-range dependencies from the future as well as from the past to be captured, which are beneficial for sequential labeling tasks. The last layer of the model is a fully connected softmax layer to allow the model to output probabilities.

3.3.1 Network Training and Regularization

The RNN network is trained using SGD for 20 epochs, using Nesterov momentum with a learning rate of 0.05 and momentum of 0.9 and a batch size of 100 sentences. The categorical cross-entropy is used as the loss function, with L2 penalty of 0.01 for regularization. The number of hidden cell units for both directions is set to 250.

Restaurant (**multi**)	
Feature	F1
Word†	.4413
+ Name List†	.5672
+ Word Cluster	.5877
+ RNN Probabilities	0.6285

Restaurant (**single**)	
Feature	F1
Word†	0.6151
+ Name List†	0.6768
+ DP Name List	0.6992
+ Word Cluster	0.7162
+ RNN Probabilities	0.7390
RNN System	0.7190

Table 2: Experimental results of 5-fold cross-validation for Slot 2. Besides using the feature stated in the current row, features stated in all previous rows are also used. **multi**: Performances of Slot 2 for Slot 1 & 2 evaluation (ignoring NULL targets). **single**: Performances of Slot 2 for Slot 2 evaluation. † indicates features used in constrained systems.

3.4 Slot 2 Features

Besides the features described in Section 2, the probability output of the RNN system is used as additional features to our CRF system. The RNN system is trained on the following input features: Word Embeddings, Name List and Word Cluster.

We performed 5-fold cross-validation experiments to obtain performances of the system after adding each feature group. Table 2 shows the experimental results. We tune the system for two different settings: Slot 2 predictions used for Slot 1 & 2 evaluation (multi setting), and Slot 2 predictions used for Slot 2 evaluation (single setting).

3.4.1 Slot 1 & 2

To generate the predictions for Slot 1 & 2 evaluation, we combine Slot 1 and Slot 2 predictions together. First, we use all Slot 2 predictions used for Slot 1 & 2 evaluation (multi setting). This covers the cases for explicit targets. To include NULL targets, we check the Slot 1 predictions for categories that are not found in the Slot 2 predictions above. These categories are assumed to belong to NULL targets.

[10] Only a single RNN model is trained with all categories collapsed into a single category.

	Slot 1									
	Restaurant					Laptop				
System	Type	Rank	P	R	F1	Type	Rank	P	R	F1
NLANGP (U)	U	1	0.7245	0.7362	0.7303	U	1	0.5685	0.4781	0.5194
NLANGP (C)	C	14	0.6454	0.6662	0.6556	C	9	0.4897	0.4468	0.4673
1st	U	1	0.7245	0.7362	0.7303	U	1	0.5685	0.4781	0.5194
2nd	U	2	0.7269	0.7308	0.7289	U	2	0.4560	0.5319	0.4910
3rd	U	3	0.7011	0.7483	0.7240	U	3	0.5000	0.4819	0.4908
Baseline	C	–	0.5419	0.6703	0.5993	C	–	0.4592	0.3166	0.3748

	Slot 2					Slot 1 & 2				
	Restaurant									
System	Type	Rank	P	R	F1	Type	Rank	P	R	F1
NLANGP (U)	U	1	0.7549	0.6944	0.7234	U	1	0.5295	0.5227	0.5261
NLANGP (C)	C	8	0.7256	0.5703	0.6386	C	3	0.4667	0.4482	0.4572
1st	U	1	0.7549	0.6944	0.7234	U	1	0.5295	0.5227	0.5261
2nd	U	2	0.7182	0.6912	0.7044	C	2	0.4901	0.4878	0.4889
3rd	U	3	0.7510	0.6062	0.6709	C	3	0.4667	0.4482	0.4572
Baseline	C	–	0.5142	0.3856	0.4407	C	–	0.3656	0.3912	0.3780

Table 3: Official results for our system, top three performing systems and baselines.

4 Results

We participated in both unconstrained and constrained settings for the English datasets. Table 3 presents the official results of our submission. For comparison, the top three performing systems and baseline results are included (Pontiki et al., 2016).

As shown from the table, our system is ranked 1st for all four evaluations we participated (Slot 1 for both restaurant and laptop domains, Slot 2 and Slot 1 & 2 for the English datasets). Similar to previous observation, the constrained systems achieved lower results than the corresponding unconstrained systems, demonstrating the use of external resources are helpful for the task.

5 Feature Ablation

The feature ablation experimental results are shown in Table 4 (Slot 1) and Table 5 (Slot 2). The neural network features contributed the most performance gains. However, using the Name List and Word Cluster features do not seem to be particularly effective on the testing data: There are negligible or negative performance gains for Slot 1. As these two features are also used in the CNN system, it may be redundant to include them again in the multi-class classification system. In addition, the neural network features may have become the dominant features during training, affecting the usefulness of other features.

Further investigation may be needed to identify better ways of combining the different machine learning systems together. For example, instead of adding neural network probability output to our multi-class classification system, we could instead add our classifier probability output as additional features to our CNN system.

6 Conclusion

In this paper, we describe our system used in classifying aspect categories (Slot 1) and extracting opinion targets (Slot 2). We explore the use of deep learning systems to provide additional neural network features to our existing system. Our system is ranked 1st in the four evaluations on the English datasets. In future, we hope to perform better feature engineering and explore how our deep learning systems can be further enhanced for the task.

Restaurant		
Feature	F1	Loss
All features	0.7303	–
- Word	0.7228	0.0075
- Head Word	0.7291	0.0012
- Name List	0.7314	-0.0011
- Word Cluster	0.7251	0.0052
- CNN Probabilities	0.6937	0.0366

Laptop		
Feature	F1	Loss
All features	0.5194	–
- Word	0.5082	0.0112
- Head Word	0.5282	-0.0088
- Word Cluster	0.5189	0.0005
- CNN Probabilities	0.4955	0.0239

Table 4: Results of ablation experiments on the testing data for Slot 1. The columns are the resulting F1 measure and F1 loss after removing a single feature group.

Restaurant		
Feature	F1	Loss
All features	0.7234	–
- Word	0.6907	0.0327
- Name List	0.6977	0.0257
- DP Name List	0.6957	0.0278
- Word Cluster	0.7086	0.0148
- RNN Probabilities	0.6813	0.0421

Table 5: Results of ablation experiments on the testing data for Slot 2. The columns are the resulting F1 measure and F1 loss after removing a single feature group.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June. Association for Computational Linguistics.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2013. A Logic Programming Approach to Aspect Extraction in Opinion Mining. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '13, pages 276–283.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, June.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, June.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27.

Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, June.

Zhiqiang Toh and Jian Su. 2015. NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, June.