Deep Learning for Aspect-Based Sentiment Analysis

Bo Wang
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
bowang@stanford.edu

Min Liu
Department of Statistics
Stanford University
Stanford, CA 94305
liumin@stanford.edu

Abstract

Sentiment analysis is an important task in natural language understanding and has a wide range of real-world applications. The typical sentiment analysis focus on predicting the positive or negative polarity of the given sentence(s). This task works in the setting that the given text has only one aspect and polarity. A more general and complicated task would be to predict the aspects mentioned in a sentence and the sentiments associated with each one of them. This generalized task is called *aspect-based sentiment analysis* (ABSA). In the annual SemEval competition, an ABSA task has been added since 2014. Among submissions of the past two years, most winning models use support vector machines (SVM).

Riding on the recent trends of deep learning, this work applies deep neural nets to solve this task. We design a combined model with aspect prediction and sentiment prediction. For both predictions, we achieve better than or close to state-of-the-art performance using deep learning models. We also propose a new method to combine the syntactic structure and convolutional neural nets to directly match aspects and corresponding polarities.

1 Introduction

Recent years has seen rapid growth of research on sentiment analysis. Sentiment analysis has both business importance and academic interest. So far, most sentiment analysis research has focused on classifying the overall sentiment of a document into positive or negative. We would, however, often like to understand what are the specific sentiments towards different aspects of an entity, e.g. a restaurant review *"Food is decent but service is so bad."* contains positive sentiment towards aspect *food* but strong negative sentiment towards aspect *service*. Classifying the overall sentiment as negative would neglect the fact that food was actually good.

In 2010, a new framework named *aspect-based sentiment analysis* (ABSA) was proposed [1] to address this problem. Here an aspect refers to an attribute or component of an entity, e.g., the screen of a cell phone, or the picture quality of a camera. An ABSA task typically involves several sub-tasks, including identifying relevant entities and aspects, determining the corresponding sentiment/polarity.

The SemEval-2015 Task 12 [2] provides a nice benchmark dataset for ABSA, which contains full reviews in laptops, restaurants and hotels. This task is otherwise similar to SemEval-2014 Task 4 except it contains full reviews instead of isolated sentences and also has an out-of-domain evaluation dataset (to be explained in Section 2).

Although deep learning models has exhibited great power in sentiment analysis [13, 15], none of the top ranking teams in SemEval-2014 Task 4 or SemEval-2015 Task 12 used deep learning models. Most teams chose SVM based algorithms [5, 6, 7, 8] or conditional random field classifiers [9, 10,

11, 12] with manually engineered features. Among all the teams accepted into the two competitions, only one team used neural networks[4] and ranked below 15th place in all subtasks.

However, we believe the inferior performance of this deep learning model does not mean deep learning should not work well with ABSA, rather, it only demonstrates a poor choice of model and training strategies. Deep neural networks have a large non-linearity and, once well trained, should be able to handle this task well. In this work, we develop a framework consisting of two deep learning models for aspect and sentiment prediction respectively. Both models outperform the best results of 2015 winning teams. The model trained on Laptop dataset also shows reasonable performance on Hotel domain.

The major contributions of this work are as follows:

1. We design a deep learning framework to extract aspects and the associated sentiments.
2. Our model achieves competitive or better performance comparing with the best results of SemEval'15 in all subtasks.
3. We propose a novel strategy to associate aspects with the corresponding sentiments.

The structure of the paper is as follows. Section 2 describes the task and datasets. In Section 3, we present detailed investigation of the training data and reasoning of our model design. Experimental results are presented in Section 4. Section 5 concludes the paper discusses future directions.

2 Problem Statement

2.1 Datasets

There are two training datasets of around 550 reviews of laptops and restaurants annotated with the corresponding aspects and polarities. In this project, we focus on the in-domain ABSA in the laptop domain and out-of-domain ABSA, so in subsequent sections subtasks only relevant to the restaurant domain are omitted. An example data snippet is included below for illustration.

```
<sentence id="B00KMRGF28_381_AH6TXTDWVUNLS:0>
  <text> Fantastic for the price, its a pity keys were not
       illuminated. </text>
  <Opinions>
    <Opinion category="LAPTOP#PRICE" polarity="positive"/>
    <Opinion category="KEYBOARD#DESIGN_FEATURES" polarity="negative"/>
  </Opinions>
</sentence>
```

2.2 Task 1: In-domain ABSA

In this task, we are given a laptop review to identify the following types of information:

Slot 1: Aspect Category (Entity and Attribute). Identify every entity E and attribute A pair E#A mentioned in the given text. E and A are chosen from predefined inventories of Entity types (e.g. keyboard, operating system) and Attribute labels (e.g. performance, design, price) per domain. The E#A inventories for the laptops domain contains 22 Entity types and 7 Attribute labels.

Slot 2: Only relevant to the restaurants domain so not described here.

Slot 3: Sentiment Polarity. Each identified E#A pair of the given text has to be assigned a polarity, from a set P = positive, negative, neutral. The neutral label applies to mildly positive or mildly negative sentiment.

2.3 Task 2: Out-of-domain ABSA

The model is tested on a previously unseen domain ("hotel"). The gold annotations for Slot 1 are provided and annotations for Slot 3 (sentiment polarity) need to be predicted.

2.4 Evaluation

The models is evaluated separately on each domain. The required output is {E#A, P} tuples for the laptops domain and polarity for the hotel domain. The prediction performance is evaluated by the F1 score in Slot 1 (Entity and Attribute) and by accuracy in Slot 3 (Polarity).

3 Model

In this section, we describe our aspect-based sentiment model. Figure 1 shows the overall architecture of the system. The system is decomposed into two parts: an aspect model and a sentiment model. The aspect model takes in a sentence vector (or a set of word vectors) and outputs a probabilistic distribution over the aspects (E#A pairs). The sentiment model takes in a sentence and outputs the sentiment of the sentence. The sentiment is connected to target aspects by augmenting the word vectors with aspect-specific re-scaling. We will describe these two models and the method to link aspects and corresponding sentiments in the following sections.

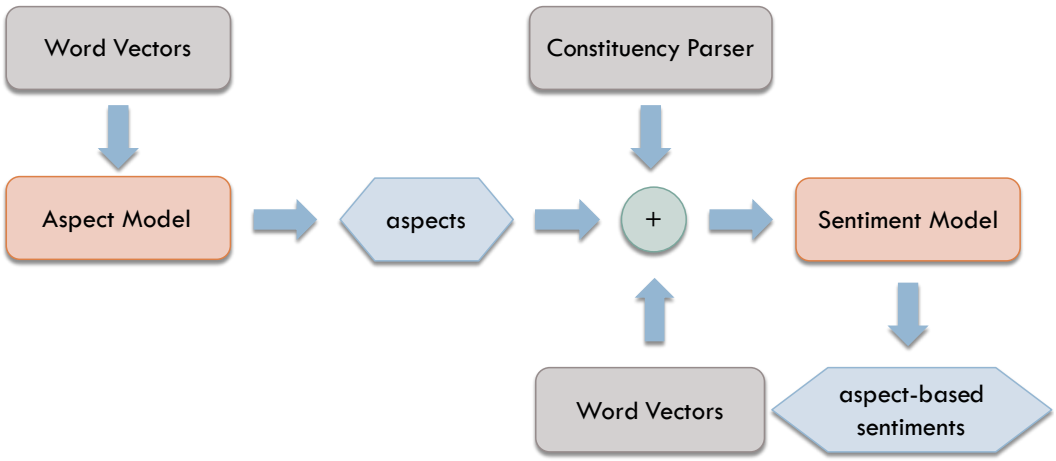


Figure 1: Overall architecture for aspect-based sentiment analysis

3.1 Aspect Model

From now on, we will use "aspect" and "E#A pair" interchangeably. Given a sentence, the aspect model predicts the E#A pairs for that sentence. This is the task of Slot 1 described in the previous section.

We adopt a two-layer neural network for this task. The first layer is fully-connected and the second layer outputs a softmax distribution. Since the training set only contains fewer than 2000 sentences, it would be difficult to train word vectors on such a small corpora. We instead use the Google News 300 dimensional word vectors[16]. Each sentence is represented by an average of the word vectors, i.e. bag-of-words model. The output of this model is a probability distribution over the 19 aspects. We use a multi-class negative entropy loss where output y is defined as $y_i = 1/k$ when the sentence has aspect i and a total of k aspects, otherwise $y_i = 0$. To make a prediction that a sentence contains aspect i, its output y_i has to exceed a threshold θ. θ is a hyperparameter selected from the validation.

The reason we did not select a more complicated model is due to the exhibited flexbility of the model. The word vectors lie in very high dimensions, a linear classifier should suffice to discriminate about a small number of classes. In fact, more complex models may result in overfitting.

One preprocessing on rare aspects is worth mentioning. There are 22 possible entities and 7 possible attributes in the laptop domain, which makes 154 total possible aspects. However, 80.7% of aspect labels in the training set belong to the 17 most frequent aspects. To predict an aspect with only a few occurances is difficult. In the preprocessing, we preserved the most common 16 aspects and combined the rest into a new aspect name "OTHER". We also introduce a "NONE" aspect which

refers to a sentence that contains no aspect. Reducing the number of possible labels from 154 to 19 also reduces model parameters by 90%, which means the model is much less prone to overfitting.

3.2 Sentiment Model

For sentiment analysis, a bag-of-words model is no longer sufficient because the interaction and location of words matter. For example, negative sentiment is often not expressed by negative words, but rather negations of positive words. A very suitable model for this task is the recursive neural tensor network (RNTN) [15], which has shown superior performance in sentiment analysis. However, our data does not have fine-grain annotation at each node of the parse tree. Thus, the inputs to RNTN are not satisfied. Instead, we adopt convolutional neural network (CNN) from [13], which only requires sentiment at the sentence level.

The CNN model applies a *filter* (or convolution) to a window of h consecutive words. The filter outputs a value after each convolution. After iterating over the whole sentence, a *feature map* is generated. Then a *max-over-time pooling* is performed over the feature map and takes the maximum value. The default setup of [13] shows very good performance on our dataset. Due to the page limit, we do not go into the details. Interested readers may refer to [13]. In the sentiment model, we still use the Google News pre-trained word vectors as the input to this model. Due to the small size of the training set, we keep the word vectors static, i.e. errors are not propagated back into the word vectors.

3.3 Aspect-based Sentiment Model

The sentiment model described in Section 3.2 is aspect-agnostic. It works fairly well with sentences of uni-sentiment. However, when judging sentences with multiple conflicting sentiments, the output is hard to predict. To solve this problem, we design the following method to connect sentiments with aspects.

Another observation is the way CNN generates responses. The convolutional layers can be viewed as weighted sum of the word vectors with respect to the shared weight matrix. Then in the max pooling layer, the largest value is selected. Thus, the magnitude of word vectors has a strong influence on the behavior of the CNN. Besides, as described in [16], all word vectors from Google word2vec are normalized to one. If a word vector is scale up (or down) uniformly on all dimensions, its impact at the max-pooling layer will be enhanced (or reduced).

The two observations above lead to our method to connect sentiments and aspects. The basic idea is to re-scale each word vector according to its relatedness to the given aspect before sending it into the CNN. Figure 2 presents the procedure using an example.

Firstly, a sentence is sent into the Aspect model which outputs the top-n aspects with respect to the specified threshold. In the example, the top-2 aspects for sentence "*Screen looks awesome; but battery is bad*" is *screen* and *battery*. For each of these top-n aspects, we filter out the probabilistic mass in that aspect. The constituency parse tree of the test sentence is generated by Stanford CoreNLP parser [17]. The tree distances between every two words are calculated using the parse tree. These distances are used in calculating the propagation of aspect-specific probabilistic mass with the following equations. $p_{i \to j}$ denotes the probability to propagate from word *i* to word *j*. $\hat{p}_j$ denotes the aggregated probability for word *j*.

$$\begin{aligned} p_{i \to j} &= p_i \cdot exp\left(-\frac{d_{ij}^2}{2h}\right) \qquad \forall i \neq j \\ p_{i \to i} &= 1 + p_i \end{aligned} \tag{1}$$

where p_i is the probability of word *i* being the given aspect; d_{ij} is the tree distance between word *i* and word *j*; h is the height of the tree and functions as a normalizer to adjust for the differences between long and short sentences. The probability of

$$\hat{p}_j = \sum_i p_{i \to j} \tag{2}$$

$\hat{p}_j$ is calculated for every word in a sentence and then re-normalized with a mean at 1. In the experiment, the typical range for $\hat{p}_j$ is tuned to between 0.7 to 1.3.

When $\hat{p}_j$ for all words in a sentence is calculated, the word vector for each word is re-scaled and then used in the Sentiment model.

$$\overrightarrow{V_j} := \hat{p}_j \overrightarrow{V_j} \tag{3}$$

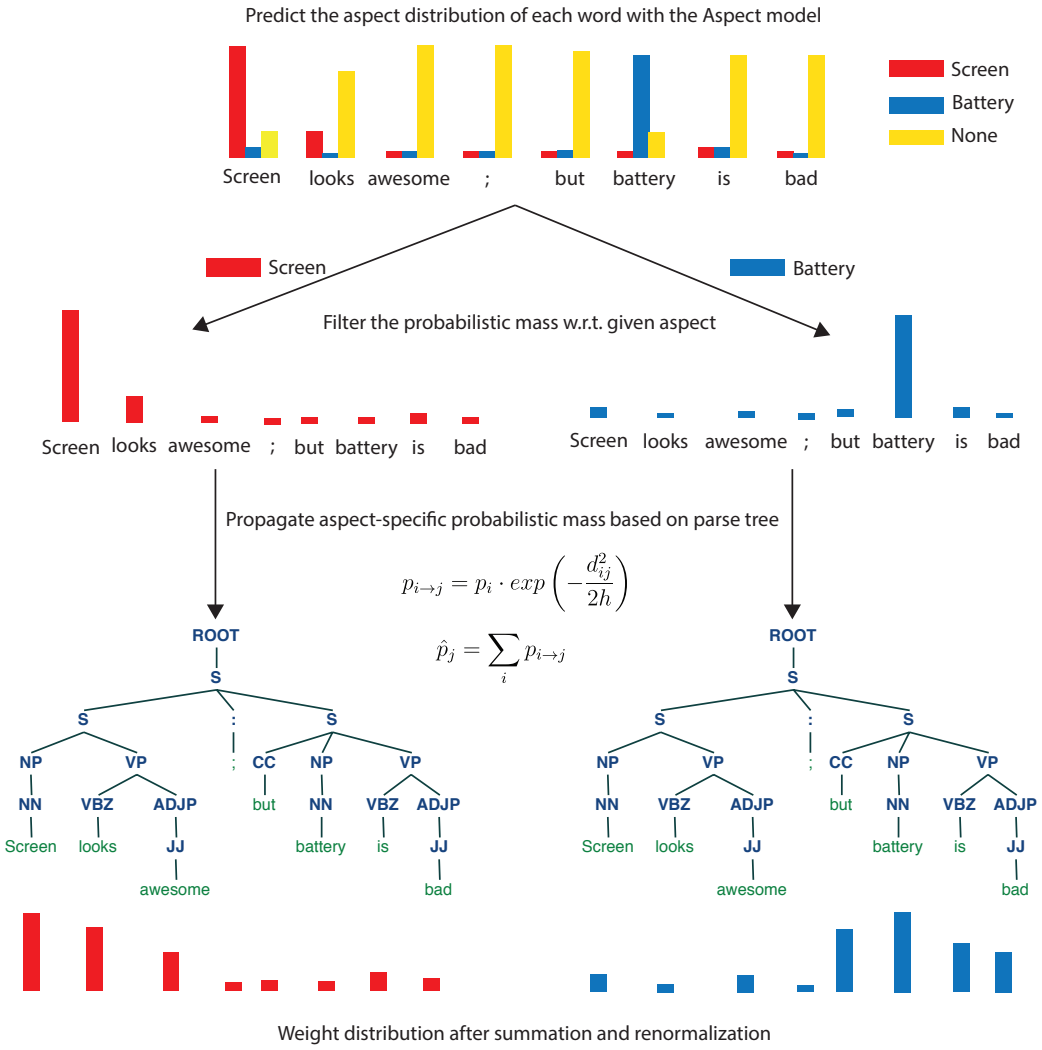


Figure 2: Aspect-based Sentiment Model

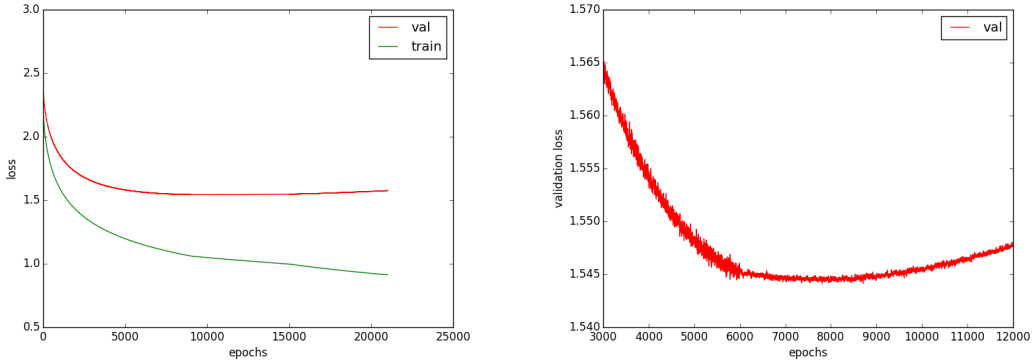
4 Experiments

4.1 Task 1 Slot 1: Aspect and Entity

In Slot 1, we randomly split the training data into a training set and a validation set to select optimal parameters. First we select the number of epochs to be the one that yields lowest validation loss, and then we select the threshold that gives best F1-measure on the validation set.

We use stochastic gradient descent to train the aspect model. The model is trained with a learning rate of 0.01 for 9000 epochs when the validation loss starts to stablize, then it is trained at learning

rate of 0.005 for 6000 epochs and the validation loss starts to curve up. Finally it is trained for another 6000 epochs at learning rate of 0.01. The plot of training and validation loss are shown in Figure 3(a). Figure 3(b) is a zoom in at the minimum of validation loss.



(a) Slot 1 Training and Validation Loss

(b) Zoom-in of Validation Loss

Figure 3: Optimal Epoch

After the optimal epoch is determined to be 7811, we sweep across the thresholds and choose optimal threshold to be 0.105 (Figure 4).

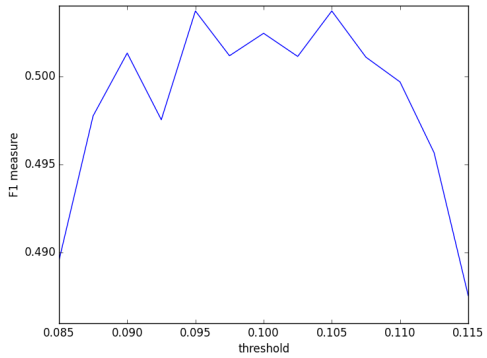


Figure 4: Optimal Threshold

We then train a model using the full training data for 7811 epochs and make predictions at the threshold 0.105 on the laptop test data. The resulting F1 measure is 0.513, which outperforms all teams of SemEval'15 competition. See Table 1 for a comparison of the performance of our model versus that of SemEval'15 Slot 1 winning team.

Model	Precision	Recall	F1 Measure
Our Aspect Model	0.526	0.501	0.513
SemEval'15 Winning Team	0.642	0.420	0.509

Table 1: Aspect Model Performance

4.2 Task 1 Slot 3: Polarity

For the sentiment model, we use ReLU nonlinearities, filter windows of 3, 4, 5 with 100 feature maps each, l_2 constraint of 3, drop-out rate of 0.5 and mini batch size of 50. The optimal training epochs is chosen by 10-fold cross validation. We use the "one standard error" rule to choose the

optimal number of epochs. The training and CV accuracy with error bars are shown in Figure 5. Epoch 6 has the highest CV accuracy, and epoch 4 is the smallest epoch number whose CV accuracy is within one standard deviation from that of epoch 6. Thus optimal number of epochs is chosen to be 4.

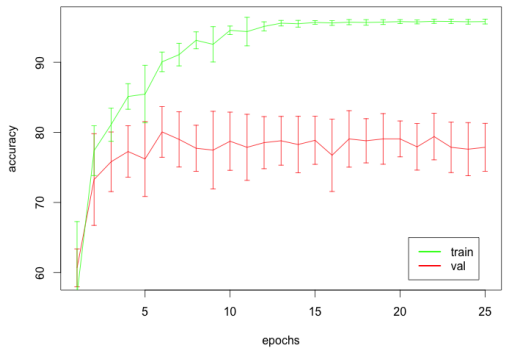
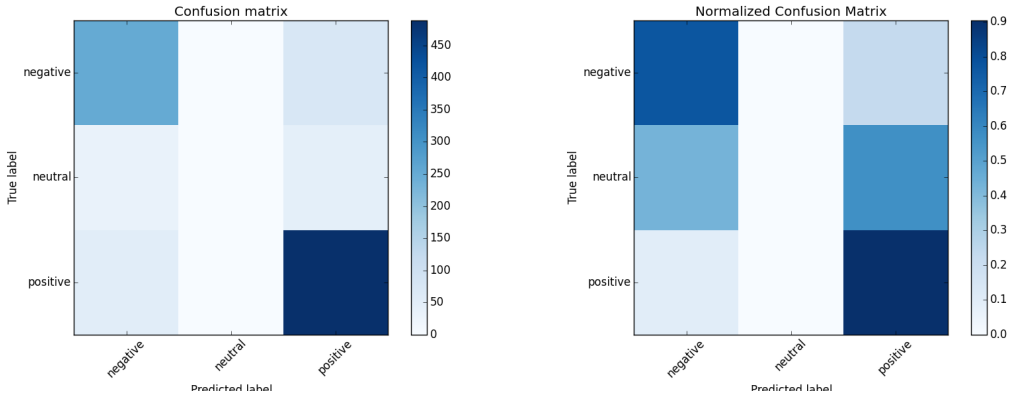


Figure 5: Optimal Epoch

We then train the sentiment model on full training data for 4 epochs and test on laptop test data. The resulting accuracy is 0.783, which is a tie with the second highest accuracy among all SemEval'15 participating teams, with the highest accuracy being 0.793.

The confusion matrix is shown in Figure 6. It can be seen that the neutral class got classified into mostly positive and negative, which is expected because only 10% of the total sentiments in training data are neutral.



(a) Confusion Matrix (b) Normalized Confusion Matrix

Figure 6: Confusion Matrix for Laptop Domain

4.3 Task 2: Out-of-domain ABSA

For this task we use unweighted word vectors as input to the sentiment model, and make predictions on sentiment. Here we cannot weight the word vectors because we are not allowed to train an aspect model on the hotel domain in advance.

Still, we got an accuracy of 0.802, which is outperformed by only two teams of the SemEval'15 competition (at accuracies 0.858 and 0.805).

We believe the prediction on hotel domain could be further improved by training the model on the restaurant dataset because laptop and hotel are very distinct domains so models trained on one

domain might not generalize to the other, while restaurant and hotel are quite similar domains so models would generalize much easily.

5 Conclusions

In this work, we design a deep-learning model to analyze the aspect-based sentiments and demonstrate competitive or better performance comparing to the best results of SemEval'15 in all subtasks. We propose a novel approach to connect sentiments with the corresponding aspects based on the constituency parse tree. This model also shows promising performance on an unseen domain. In the future work, we are interested to test the model on other (larger) datasets and evaluate the performance of transfer learning. We would also like to explore more sophisticated models in aspect prediction by using adaptive thresholds.

References

[1] Thet, T. T. & Na J. C. & Khoo C.S.G (2010) Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science archive Volume 36 Issue 6* pp. 823-848 Sage Publications, Inc. Thousand Oaks, CA, USA

[2] Androutsopoulos I. et al. (2015) SemEval-2015 Task 12: Aspect Based Sentiment Analysis, *SemEval 2015, International Workshop on Sementic Evaluation*

[3] Pontiki M. et al. (2014) SemEval-2014 Task 4: Aspect Based Sentiment Analysis, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 27 - 35, Dublin, Ireland, August 23-24, 2014.

[4] Blinov P. & Kotelnikov E. (2014) Blinov: Distributed Representations of Words for Aspect-Based Sentiment Analysis at SemEval 2014, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 140 - 144, Dublin, Ireland, August 23-24, 2014.

[5] Wagner J. et al. (2014) DCU: Aspect-based Polarity Classification for SemEval Task 4, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 223 - 229, Dublin, Ireland, August 23-24, 2014.

[6] Kiritchenko S. et al. (2014) NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 437 - 442, Dublin, Ireland, August 23-24, 2014.

[7] Brychcin T. et al. (2014) UWB: Machine Learning Approach to Aspect-Based Sentiment Analysis, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 817 - 822, Dublin, Ireland, August 23-24, 2014.

[8] Brun C. et al. (2014) XRCE: Hybrid Classification for Aspect-based Sentiment Analysis, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 838 - 842, Dublin, Ireland, August 23-24, 2014.

[9] Toh Z. & Wang W. (2014) DLIREC: Aspect Term Extraction and Term Polarity Classification System, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pp. 235 - 240, Dublin, Ireland, August 23-24, 2014.

[10] De Clercq O. et al. (2015) LT3: Applying Hybrid Terminology Extraction to Aspect-Based Sentiment Analysis, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.

[11] Toh Z. & Su J. (2015) NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.

[12] Hamdan H. et al. (2015) Lsislif: CRF and Logistic Regression for Opinion Target Extraction andSentiment Polarity Analysis, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.

[13] Kim Y. (2014) Convolutional Neural Networks for Sentence Classification, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* pp. 17461751, Doha, Qatar, October 25-29, 2014

[14] Pontiki M. et al. (2014) SemEval-2015 Task 12: Aspect Based Sentiment Analysis, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.

[15] Socher R. et al. (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*

[16] Mikolov T. et al. (2013) Distributed Representations of Words and Phrases and their Compositionality, *Conference on Neural Information Processing Systems (NIPS2013)*

[17] Manning, Christopher D., et al (2014) The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.