NileTMRG at SemEval-2016 Task 5: Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction

Talaat Khalil and **Samhaa R. El-Beltagy**
Center for informatics sciences
Nile University
Giza, Egypt
t.maher@nu.edu.eg, samhaa@computer.org

Abstract

This paper describes our participation in the SemEval-2016 task 5, Aspect Based Sentiment Analysis (ABSA). We participated in two slots in the sentence level ABSA (Subtask 1) namely: aspect category extraction (Slot 1) and sentiment polarity extraction (Slot 3) in English Restaurants and Laptops reviews. For Slot 1, we applied different models for each domain. In the restaurants domain, we used an ensemble classifier for each aspect which is a combination of a Convolutional Neural Network (CNN) classifier initialized with pre-trained word vectors, and a Support Vector Machine (SVM) classifier based on the bag of words model. For the Laptops domain, we used only one CNN classifier that predicts the aspects based on a probability threshold. For Slot 3, we incorporated domain and aspect knowledge in one ensemble CNN classifier initialized with fine-tuned word vectors and used it in both domains. In the Restaurants domain, our system achieved the 2nd and the 3rd places in Slot 1 and Slot 3 respectively. However, we ranked the 8th in Slot 1 and the 5th in Slot 3 in the Laptops domain. Our extended experiments show our system could have ranked 2nd in the Laptops domain in Slot 1 and Slot 3, had we followed the same approach we followed in the Restaurants domain in slot 1 and trained each domain separately in Slot 3.

1 Introduction

Due to the increasing numbers of user generated reviews written every day within e-commerce websites, a great interest has been shown in the sentiment analysis research community to build intelligent systems that can accurately tackle the task of sentiment analysis in these reviews.

In this context, the SemEval-2016 ABSA, task 5[1], Subtask 1 addresses a number of research problems related to this topic, including building systems that are able to extract aspect categories (Slot 1) and determine the sentiment polarity towards each aspect in each sentence (Slot-3) which were the two slots in which we participated.

The best results for Slot 1 in SemEval-2015 (Pontiki et al., 2015), were achieved by the NLANGP team (Toh and Su, 2015). The team tackled the problem by modeling it as a multi-class classification problem with binary classifiers for each aspect. They used a neural network with one hidden layer and features based on word n-grams, brown and k-means word clusters from Amazon and Yelp datasets and parsing features. For Slot 3, the best results were achieved by the Sentiue team (Saias, 2015) who used a Maximum Entropy classifier with domain and aspect features and features based on word n-grams, lemmas, negation terms, exclamation and question marks, sentiment lexicons, and POS tags.

[1] http://alt.qcri.org/semeval2016/task5/

Proceedings of SemEval-2016, pages 271–276,
San Diego, California, June 16-17, 2016.

This year, when addressing Slot 1, we participated with a system that can extract aspects in English reviews in the two domains that the task provided test sets for, namely: restaurants (REST) and laptops (LAPT). For the restaurants domain we treated the problem as a multi-class classification problem using an ensemble binary classifier for each aspect which is a combination of a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) classifier and a Convolutional Neural Network (CNN). While the SVM classifier features were based on a Bag of words model, the CNN classifier was initialized with pre-trained word vectors based on the architecture proposed by Kim (2014). For the Laptops domain, we used one CNN classifier that outputs probability scores for each aspect, then a threshold was applied so that only outputs with scores higher than that threshold were predicted as aspects.

For Slot 3, we incorporated domain and aspect information in one ensemble classifier consisting of three CNNs trained using the whole training data provided in both domains and initialized with word vectors that were fine-tuned using training examples collected in a semi supervised way by the same CNN architecture as in an earlier phase.

The rest of this paper is organized as follows: section 2, describes the system architecture and settings, while section 3, presents and discusses our system performance and evaluation. Finally, section 4 concludes the paper and presents some ideas for potential future work.

2 System Details

To train our models, we depended mainly on the official training data provided by the SemEval-2016 task organizers. Furthermore, for choosing the best parameters and architectures, we used the SemEval-2015 (Pontiki, et al., 2015) training set for training and SemEval-2015 test set as a validation set. The validation set was used to choose the best model for each domain and to tune the network hyperparameters. In aspect category extraction (Slot 1), we used only the training data, however we considered our submission as an unconstrained one because we initialized our models using pre-trained publicly available word vectors[2] trained on a subset of Google news using the word2vec model (Mikolov, et al., 2013). For polarity extraction (Slot 3), we used additional external examples for training and fine-tuning the word vectors from the Yelp Academic Dataset[3] and Amazon electronics reviews (Jo, 2011).

In the next subsections, we discuss the used CNN architecture as it's involved in all of our models. Then we discuss each model in detail.

2.1 Convolutional Neural Network architecture

Our CNN implementation is based on the architecture proposed by Kim (2014). In this model, each sentence is represented as a concatenation of all its word vectors which can be described using Equation (1),

$$\boldsymbol{x}_{1:n} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \dots \boldsymbol{x}_n \quad (1)$$

where $\boldsymbol{x}_i \in \mathbb{R}^k$ is a k dimensional word vector for the i^{th} word in the sentence, $\oplus$ is the concatenation operator, and $\boldsymbol{x}_{1:n}$ is the model input vector, which is the concatenated word vectors of the sentence from the first word to the n^{th} word, where n is the number of words in the sentence.

A convolution filter $\boldsymbol{w}$ of width h words is then applied to the input vector to produce new features by simply taking the dot product between the filter and the corresponding input vector slice, then adding a bias factor b, and finally applying the non-linearity function f. The filter is then shifted by one word and applied again to produce the next feature until we reach the end of the input vector. This operation is clarified in Equation (2),

$$\boldsymbol{c}_i = f(\boldsymbol{w} \cdot \boldsymbol{x}_{i:i+h-1} + b) \quad (2)$$

where $\boldsymbol{c}_i$ is the generated feature resulting from applying the filter w on the input vector slice from the start of word i to the end of word $i + h - 1$.

As a result of the convolution operation, a feature map $\boldsymbol{c} \in \mathbb{R}^{n-h+1}$ is generated where:

$$\boldsymbol{c} = [\boldsymbol{c_1}, \boldsymbol{c_2}, \dots, \boldsymbol{c_{n-h+1}}] \quad (3)$$

A max-pooling operation (Collobert et al., 2011) is then applied to the feature map vector $\boldsymbol{c}$ by only taking the maximum value $\hat{c} = \max\{\boldsymbol{c}\}$ and considering it as a hidden layer feature generated from the corresponding convolution filter.

Since one feature is not enough to represent the needed knowledge, the process is repeated several

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

[3] https://www.yelp.com/dataset_challenge/

times using different filters and different filter widths. These features form a hidden layer, which is followed by a Softmax layer that outputs prediction probabilities for each output class.

To prevent overfitting to the training data, we employed 'dropout' on the hidden layer and constrained its weights by l_2-norms (Hinton, et al., 2012). This was done by randomly dropping some of the hidden layer units by a probability rate p while training to prevent over adaptation to certain units. We also applied a constraint on the l_2-norms, by rescaling the weights connecting the hidden layer and the output layer such that they are limited by an upper limit s after each update step.

For training, the backpropagation algorithm (Rumelhart, et al., 1986) was applied and the network weights were updated using mini-batch stochastic gradient descent with the Adadelta update rule (Zeiler, 2012), which considers a separate adaptive update rate for each weight. As will be detailed in the following subsections, we have mostly employed Static-CNN, where initialized input vectors are kept as is. However, there were cases where we also employed Dynamic-CNN. In Dynamic-CNN, input vectors are updated for optimizing the network.

For choosing the CNN hyperparameters, we started by using the ones which were validated by Kim (2014) and then we tried different values on our validation set. We ended up by choosing convolution filter window widths (h) of 3, 4, and 5 with 100 feature maps for each width. We used the rectified linear units as a non-linearity activation function, which simply outputs the maximum of zero and the input value. We set the dropout rate (p) to 0.5 and the l_2 maximum value (s) to 3. We set the number of optimization iterations over the whole data (epochs) and the mini-batch size to 25 except for the official Laptops aspect extraction model (LAEM), where the values were set to 100 and 50 accordingly.

2.2 Restaurants Aspect Extraction Model (RAEM)

To extract aspect categories for the restaurants domain, we dealt with the problem as a multi-class classification problem where we have a binary ensemble classifier for each of the 12 aspects, trained on the aspect data against all other aspects' data (one vs all strategy).

This ensemble classifier is a combination of two classifiers; one Static-CNN classifier initialized using the Google news word vectors, and one SVM classifier which was trained using word unigram counts weighted by the inverse word frequencies in the training data (IDF) (Salton and Buckley, 1988). Vectors for words that had no corresponding entry in the Google news vectors, were set to zeros. The model predicts the aspect if any of the two classifiers predicts it.

The SVM classifier had a high precision on the validation set but a very low recall. Using it as part of our ensemble thus increased the total F-Score on the validation set.

2.3 Laptops Aspect Extraction Model (LAEM)

To extract aspects from laptop reviews, we used a Static-CNN classifier with 81 output nodes representing all the aspects found in the Laptops training set. At test time, the classifier predicts the aspect if its output nodes' probability score exceeds a certain threshold.

To choose the best threshold, we tried different values on our validation set and found out that a threshold value of 0.18 performed best, followed by 0.16 which was slightly less in terms of the total F-Score. We preferred to use 0.16 as a threshold to prevent choosing a value resulting from overfitting on the validation set.

We chose to use this model for the laptops domain as using the alternative one vs all strategy needed 81 classifiers to be trained which is computationally slower during training and testing. The one classifier model also performed slightly better on the validation set at the chosen threshold. As we show in the extended experiments section (section 3.2), this was not the best strategy in approaching this problem.

2.4 The Sentiment Extraction Model (SEM)

For Slot 3, an ensemble model was used for both domains. This model counts votes from three classifiers and predicts the class which has the maximum number of votes from the three classes namely: the positive (Pos), the negative (Neg), and the neutral (Neu).

Data Domain	Positive Sentences	Negative Sentences
Amazon Dataset	50,000	18,326
Yelp Restaurants	50,000	50,000
Yelp Computers	1974	2230
Yelp Electronics	3478	2226

Table 1: Fine-tuning data distribution.

Data Domain	Neutral Example	Negative Examples
Amazon Dataset	22	2
Yelp Restaurants Dataset	78	24

Table 2: Hand labeled data distribution.

We adopted this voting criterion as it yielded slightly better results on the validation set. Each one of the three classifiers is similar to the one which was discussed in section 2.1 with a slight variation resulting from incorporating domain and aspect knowledge into the CNN model. This incorporation was done by introducing new binary features to the hidden layer of the CNN. The new features indicate the presence or the absence of a certain aspect or domain in a given sentence.

The problem with a word vector is that it represents a word by its semantic meaning captured through its context while the sentiments are not captured directly. To tackle this problem, we trained a Dynamic-CNN on sentences tokenized from the Yelp academic dataset reviews in restaurants, computers and electronics domains as well as data obtained from electronics and laptop reviews from Amazon. We employed a distant supervision method where five star review sentences are considered to be positive, while the one star ones are considered to be negative. We could not apply the same method for getting neutral sentences because we noticed that the three stars reviews can be mainly a combination of positive and negative sentences rather than neutral ones. Using these reviews could simply introduce more noise. A number of sentences were then sampled randomly from these reviews and used for fine-tuning the word vectors. Domain features were added when possible, otherwise they were set to zeros. The distribution of this collected dataset over the polarity labels and domains is clarified in Table 1.

Model	Domain	P	R	F
RAEM (Official)	REST	72.69	73.08	72.88
RAEM without SVM	REST	76.76	68.91	72.62
LAEM (Official)	LAPT	44.25	50.56	47.19
RAEM	LAPT	59.44	45.3	51.42
RAEM without SVM	LAPT	63.27	39.67	48.76

Table 3: Results for Slot 1 in terms of precision, recall, and F-Score.

Model	Domain	Pos-F	Neg-F	Neu-F	Acc.
SEM (Official)	REST	91.63	75.44	32.73	85.44
Domain Specific SEM	REST	91.25	74.94	32.73	85.09
SEM (Official)	LAPT	83.55	72.95	8.00	77.40
Domain Specific SEM	LAPT	84.85	73.65	11.76	78.65

Table 4: Results for Slot 3 in terms of positive, negative, neutral F-Score and accuracy.

After tuning the word vectors, three Static-CNNs with incorporated domain and aspect features were initialized with them, with random weights initializations, and trained on the whole training data from both domains in addition to 310 hand labeled examples which we added as an attempt to balance the training set label distribution across the two tackled domains (REST and LAPT) and to increase the number of the neutral labels as there were very few of them in the official training set compared to the other two polarities. The distribution of the hand labeled examples is shown in Table 2.

3 Evaluation and Results

In this section we discuss our official scores in the SemEval-2016 ABSA task. Furthermore, we present other experiments that were not officially submitted, but which provide insights regarding the adopted models as well as better performance for the laptops domain, than what was submitted.

3.1 Official Participations

For the aspect category extraction task (Slot 1), our RAEM model achieved the 2nd place out of 30 teams in the restaurants domain, with an F-Score of 72.886 which is only 0.145 less than what was achieved by the best performer. In the laptops domain, our LAEM model ranked the 8th amongst the 22 participating teams with an F-Score of 47.196. The detailed results in terms of precession, recall, and F-Score are shown in Table 3.

For the sentiment extraction task (Slot 3), we ranked 3rd and 5th in the restaurants and the laptops domains with accuracy scores of 85.448 and 77.403 respectively. The official evaluation accuracies and the per-class F-Scores are shown in Table 4. It can be deduced from the F-Scores that the number of training examples per class matters as the positive F-Score is always the best, followed by the negative and the neutral is always the worst which is a real reflection of the bias of the official training data distribution and the fine-tuning data.

3.2 Extended Experiments

In addition to the official submissions, we ran some additional experiments that were evaluated using the official scripts provided by the task organizers.

For Slot 1, given that the RAEM model achieved a good result in the restaurants domain, we decided to train this model on the laptops data. The resulting model achieved an F-score of 51.42 which would have put the system in 2nd place. We also experimented using the RAEM model without using the SVM classifier and reported that in Table 3 which shows that it contributed to enhancing the recall especially in the laptops domain. However, the RAEM system on its own, seems to perform relatively well even without the help of the SVM.

For Slot 3, we conducted two other experiments in which we used the fine-tuned word vectors to initialize two different ensemble classifiers like the one which was described in the SEM. However, here we separated the data so that we there is one classifier per domain. This provided better results in the laptops domain which would have ranked as the 2nd best performer, but decreased the accuracy slightly in the restaurants domain as shown in Table 4 as Domain Specific SEM.

4 Conclusions

In this paper, we presented models for extracting aspects and their corresponding sentiment polarities from user reviews in SemEval-2016, task 5. The proposed models achieved comparable scores to state of the art results on the test set without any feature engineering efforts.

Our experiments show that the best performance in the aspect category extraction task can be achieved by using one binary classifier per aspect following the one vs all strategy. For the sentiment extraction task, our results show that after fine tuning the word vectors, it is better to train a separate classifier for each domain.

We believe that bigger and more class balanced training and fine-tuning datasets can boost the results as it was clear that the classes' distributions are reflected in the testing results. We used fine-tuned word vectors for training the CNN model used for sentiment determination, initializing its weights randomly. In the future we plan investigating using the weights of the CNN that was used to fine tune the word vectors to initialize the second CNN used for sentiment classification. We expect that this might have a positive impact on the performance of the classifier, but this remains to be tested.

References

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537. http://doi.org/10.1145/2347736.2347755

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*, 273–297.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv: 1207.0580*, 1–18. http://doi.org/arXiv:1207.0580

Jo, Y. (2011). Aspect and Sentiment Unification Model for Online Review Analysis. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 815–824. http://doi.org/10.1145/1935826.1935932

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 1746–1751. Retrieved from

http://emnlp2014.org/papers/pdf/EMNLP2014181.pdf
Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and I. A. (2015). SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado.* (pp. 486–495). Denver, Colorado.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality.
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. http://doi.org/10.1038/323533a0
Saias, J. (2015). Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 12. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 767–771). Denver, Colorado.
Salton, G., & Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, *24*(5), 513–523.
Toh, Z., & Su, J. (2015). NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 496–501).
Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *arXiv*, 6. Retrieved from http://arxiv.org/abs/1212.5701