DOI: http://doi.org/10.4038/icter.v12i1.7201

A Hybrid Approach for Aspect Extraction from Customer Reviews

Yasas Senarath[#1], Nadheesh Jihan[#2], Surangika Ranathunga[#3]

Abstract— **Aspect Extraction from consumer reviews has become an essential factor for successful Aspect Based Sentiment Analysis. Typical user trends to mention his opinion against several aspects in a single review; therefore, aspect extraction has been tackled as a multi-label classification task. Due to its complexity and the variety across different domains, yet, no single system has been able to achieve comparable accuracy levels to the human-accuracy. However, novel neural network architectures and hybrid approaches have shown promising results for aspect extraction. (Support Vector Machines) SVMs and (Convolutional Neural Networks) CNNs pose a viable solution to the multi-label text classification task and has been successfully applied to identify aspects in reviews. In this paper, we first define an improved CNN architecture for aspect extraction which achieves comparable results against the current state-of-the-art systems. Then we propose a mixture of classifiers for aspect extraction, combining the proposed improved CNN with an SVM that uses the state-of-the-art manually engineered features. The combined system outperforms the results of individual systems while showing a significant improvement over the state-of-the-art aspect extraction systems that employ complex neural architectures such as MTNA.**

Keywords— **Aspect Extraction, Deep Learning, Sentiment Analysis, Text Classification, Natural Language Processing**

I. INTRODUCTION

Customer reviews have become the means of expressing opinions and views of consumers towards different aspects of products and services. The information contained in such reviews can be leveraged by customers to identify the best available products/ services in the market and by the organizations to identify and satisfy customer needs. However, customer reviews are in unstructured textual form, which makes it difficult to be summarized by a computer. In addition, manual analysis of this huge amount of data for information extraction is nearly impossible. Automatic sentiment analysis of customer reviews has, therefore, become a priority for the research community in recent years.

Conventional sentiment analysis of text focuses on the opinion of the entire text or the sentence. In the case of consumer reviews, it has been observed that customers often talk about multiple aspects of an entity and express an opinion on each aspect separately rather than expressing the opinion towards the entity. Aspect Based Sentiment Analysis (ABSA) has emerged to tackle this issue.

The goal of Aspect Based Sentiment Analysis is to identify aspects present in the text, and the opinions expressed for each aspect [1]. One of the most important tasks of ABSA is to extract aspects from the review text.

However, there have been several challenges in extracting aspects such as support for multiple domains, detecting multiple aspects in a single sentence, and detecting implicit aspects [2]. State-of-the-art systems presented by Kim et al. [3] and Jihan et al. [4] try to address the above challenges, but those systems lack in terms of performance.

Moreover, Neural Network models have increasingly been used in text classification and aspect extraction [5, 6, 7]. Among these Neural Network models, a common type is the Convolutional Neural Networks (CNN) [3, 5]. However, existing state-of-the-art CNN architectures used in text classification for aspect extraction do not incorporate improvements [7, 8] (e.g. non-static CNN, multi-kernel convolution layers, and optimizing the number of hidden layers and hidden neurons) that have been identified as beneficial for general text classification tasks [3]. Moreover, traditional CNN models lack the ability to capture context level features. There have been models based on CNNs used to extract aspects from customer reviews [5, 6].

In the light of the above identified limitations of traditional CNN models for aspect extraction, this paper presents following contributions:

- We present a modified CNN architecture for aspect extraction, which implements two improvements. To capture context level features, we incorporate multiple convolutional kernels with different filter sizes. We also introduce dropout regularization to prevent models from over-fitting to the training samples. Although these improvements have been used in general text classification tasks [3], the effect of the same has not been explored for aspect extraction.
- We implement an optimal dense layer architecture between the feature selection layer and the output layer of the CNN with the use of a feed-forward network with two hidden layers that was derived using the constructive method proposed by Huang et al. [7]. This also helps to calculate the optimal number of hidden neurons for each layer that is sufficient to store the relationship between the training instances and the classes. The effect of such optimization techniques on hidden dense layers of the CNN models is not yet investigated for aspect extraction or text classification tasks.
- We compare the effect of initiating the word embedding features for CNN models using Skip-gram and Continuous bag of word (CBOW) trained word2vec [8] models for aspect extraction. Although

Manuscript received on 4 March 2019. Recommended by Prof. G.K.A. Dias on 10 June 2019.

This paper is an extended version of the paper "Aspect Extraction from Customer Reviews Using Convolutional Neural Networks" presented at the ICTer 2018.

Yasas Senarath, Nadheesh Jihan and Surangika Ranathunga are from Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. (*wayasas.13@cse.mrt.ac.lk*, *nadheeshj.13@cse.mrt.ac.lk*, *surangika@cse.mrt.ac.lk*)

related research reported the use of CBOW models for aspect extraction, the optimal technique for the same has not been identified through a comparative study.

- We show that the use of non-static CNN models (that update word vectors during training) perform better than static models (that do not update word vectors during training) for aspect extraction, in the absence of word2vec models trained with domain-specific corpora.
- We incorporate prediction probabilities from SVM aspect classification model [4] to improve the performance of our CNN with the expectation that manually constructed features could help to improve the overall performance.

The SemEval Task 5 datasets [9] for Restaurant and Laptop domain have been used in this research for training and evaluation of the models. We were able to significantly outperform the current state-of-the-art techniques for multi-domain aspect extraction using our mixture of classifier.

The rest of the paper is organized as follows. In section 2, related work is discussed. Section 3 explains the SemEval-2016 Task 5 dataset. Section 4 elaborates our aspect classifier models in detail. Experimental results are discussed in section 5. Finally, section 6 concludes the paper.

II. Literature Review

In the recent literature, majority of work on aspect detection is performed using supervised and hybrid machine learning approaches. Machacek [10] presented a supervised machine learning approach using bigram bag of words model. Although this model was tuned with several different features extracted manually, it has not represented the sentence well as opposed to CNN models that capture features automatically during training.

In contrast to the traditional supervised machine learning methods, Toh et al. [5] presented a hybrid approach, which uses a CNN along with a binary classifier. This system was the top ranked system in the SemEval 2016 Task 5 competition. Furthermore, Khalil and El-Beltagy [11] used an ensemble classifier that used a combination of a CNN initialized with pre-trained word vectors and a Support Vector Machines (SVM) classifier with a bag of words model as features.

It has also been shown that CNN architecture performs well in multiple other text categorization tasks [3]. Kim [3] has experimented with a CNN model with static and non-static channels of word vectors to represent a sentence. He has observed that non-static CNN has outperformed static CNN for a significant number of datasets. However, these experiments have not been carried out for aspect extraction.

Jihan et al. [4] use an SVM to predict the aspect category with multiple features extracted from text. They have used a clever pre-processing pipeline to clean and normalize text data. This model has obtained a F1 score of 74.18 and 52.21 for datasets from restaurant and laptop domains (respectively) provided in SemEval-2016 task 5. Furthermore, MTNA [12] obtained a F1 score of 76.42 on the restaurant dataset by training a set of one-vs-all deep neural network models consisting of an LSTM layer followed by a CNN layer using both aspect category and aspect term information. We consider these two systems as our benchmark.

III. SemEval-2016 Task 5 dataset

Existence of a dataset such as the one provided by SemEval-2016 task 5 provides a standardized evaluation technique to publish our results, and they can be compared fairly with other systems, which are evaluated on the same dataset. Previously many different researchers used various data sets in their publications, making it difficult to compare the results obtained.

Our proposed CNN classifier and the baseline CNN are trained using the official SemEval-2016 Task 5 dataset of reviews for restaurant (training: 2000, testing 676 sentences) and laptop (training: 2500, testing 808 sentences) domains. Training sentences are annotated for opinions with respective aspect category, while taking the context of the whole review into consideration. Sentences are classified under 12 and 81 classes in the restaurant and laptop domains, respectively.

IV. Methodology

This section describes the architectures for the mixture of classifiers that we propose for the task of aspect extraction. Convolutional Neural Network architecture is presented in Section A, in Section B we introduce word2vec embedding, Support Vector Machine classifier and features used are introduced in Section C, and proposed mixture of classifiers in Section D.

A. Convolutional Neural Network

Our CNN model is inspired by the text classification architecture proposed by Kim [3], and the work done by Toh et al. [5] for aspect extraction.

In implementing the CNN, each sentence is represented with a $R^{n \times k}$ sentence feature matrix, where each row is the feature vector of the corresponding word. Here n is the number of words in the sentence, and k is the size of the feature vector. We only used word vectors for each word as the features. Even though the convolutional layer requires a sentence matrix with a fixed size, customer reviews have different word counts. Therefore, a padding tag was added to extend the sentence length to a predefined length, thus allowing all the sentences to have the same length.

1) *Baseline CNN:* Our baseline CNN is similar to the CNN presented by Toh et al. *[5]*. In this model, a convolution layer with a window size of w is applied to the sentence feature matrix to generate new features. We use zero padding for convolutional operations to generate a feature map with the same height as the sentence matrix. Then the max pooling layer is applied to select the most important feature from each feature map. Then we use a single hidden dense layer as proposed by Toh et al. *[5]*.

Using the output from the last dense layer, the Softmax layer computes the probabilities of having each aspect in each sentence. Then a predefined threshold value (th) is used to classify each sentence to the aspect categories according to the probability outputs from the Softmax layer. Toh et al. *[5]* introduced another category for sentences with no aspects. However, we consider this as redundant. We can determine the sentences without any aspects when all the probability values for each aspect are less than the threshold.

2) *Improved CNN:* CNN model used by Toh et al. *[5]* contains a convolution layer with a single kernel. Since the convolutional kernel has a fixed window size, determining that value to capture most of the contextual information is a difficult task. With a small kernel, the convolutional layer may fail to capture contextual information and semantic relationships that are larger than the selected kernel size. Choosing a very large kernel can degrade the quality of features by capturing multiple contextual information into a single feature. Therefore, the convolution layer of our improved CNN uses several convolutional kernels with different filter sizes and single step stride, thus generating a $1 \times n$ feature map for each filter. Use of the convolutional layer with multiple kernel sizes provides more flexibility to the CNN model to extract semantic relationships with various lengths as the features.

Toh et al. *[5]* used only a single hidden dense layer with Rectified Linear Unit (ReLU) activation. However, Huang et al. *[7]* constructively proved that a two-hidden layer feed-forward networks with $2\sqrt{(m+2)N}(\ll N)$ hidden units can be used to learn N distinct samples with any arbitrarily small error, where m is the number of output neurons. If we consider the outputs from the convolutional layer as features and the Softmax layers as the output layer with m number of hidden units, then we can implement the two hidden layer feed-forward network in between those two layers replacing the single hidden layer in the baseline CNN. Therefore, we introduced two hidden layers L_1 and L_2 withh_1 and h_2 hidden units, respectively. The hidden units h_1 and h_2 are determined using equations (1) and (2) as proposed by Huang et al. *[7]*.

$$\mathrm{h}_1 = \sqrt{\mathrm{N} \times (\mathrm{m}+2)} + 2 \times \sqrt{\mathrm{N}/(\mathrm{m}+2)} \quad (1)$$

$$h_2 = m \times \sqrt{N/(m+2)} \quad (2)$$

Kim [3] shows that using dropout to prevent co-adaptation of hidden units by randomly dropping a proportion of hidden units can significantly improve the CNN for general sentence classification tasks. Therefore, we introduced a dropout layer instead of kernel regularization to our CNN implementation to perform dropout regularization [13] to prevent the model from over-fitting to the training data.

Figure 1 shows the network structure of our improved CNN. It presents the process of extracting convolutional features from the sentence matrix using two convolutional kernels. Then the max pooling layer selects the best features from both convolutional feature matrices extracted by two convolutional kernels. The output neurons from max-pooling layers are transformed to class probability outputs using the two-hidden layers and the Softmax layer.

B. *Word2Vec Embedding*

Mikolov et al. [8] presented CBOW and Skip-gram architectures to implement word2vec models. The CBOW architecture predicts the current word based on the context (surrounding words), whereas the Skip-gram architectures use the current word to predict the surrounding words (context) [8]. Kim [3] showed that in the absence of a large supervised training set, initializing the feature vector using word2vec improves the performance of the CNN model for text classification tasks. Even though Toh et al. [5] and Khalil et al. [11] have only used the CBOW trained word2vec models to train CNN models for aspect extraction, a comparative study of the performance of CBOW and Skip-gram to initiate word embeddings to train CNN models for text classification is not available.

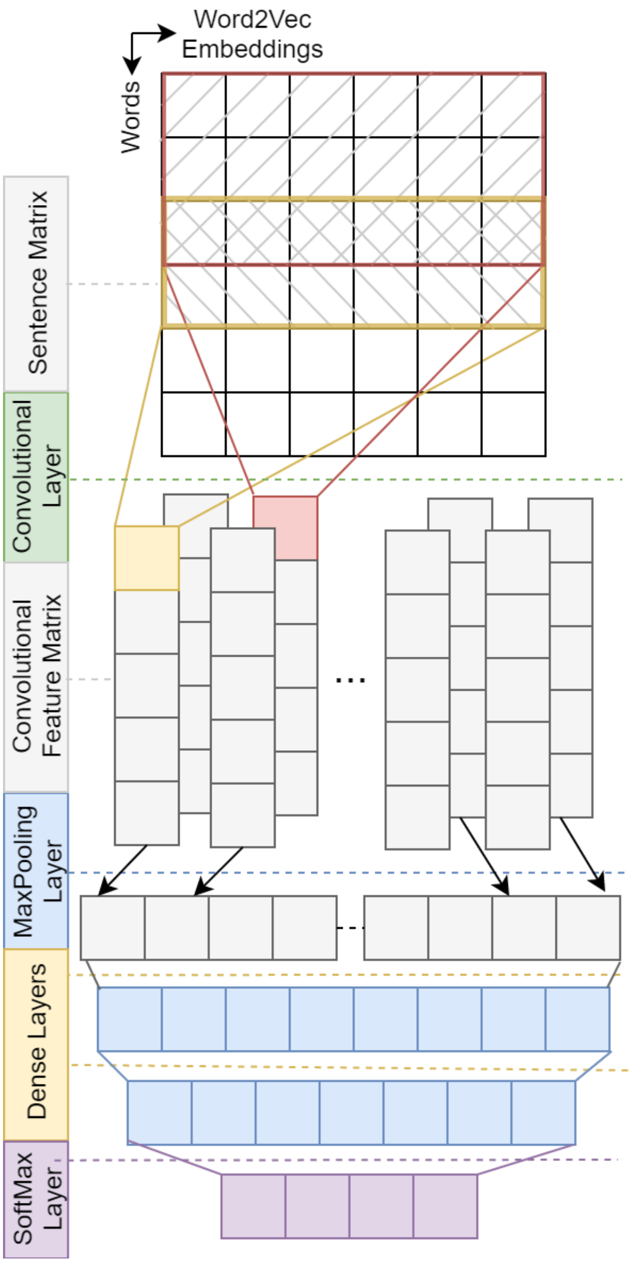

Fig. 1 The architecture of our Convolutional Neural Network

Thus, we tried both Continuous Bag of Words (CBOW) and Skip-gram trained word2Vec models to initiate word embedding features for the improved CNN model. The word2Vec models were trained using the Yelp[1] and Amazon product review[2] datasets. In addition, we trained both the CNN models with Google's pre-trained word2vec (CBOW trained)

[1] https://www.yelp.com/dataset/_challenge

[2] http://jmcauley.ucsd.edu/data/amazon/

TABLE I
HYPER-PARAMETERS OF BASELINE CNN

Layer	Parameter	Value
Convolutional Layer	Window size	5
	# of filters	300
	Activation	tanh
Hidden Layer	# of Neurons	100
	Activation	ReLU
Training parameters	Batch size	50
	Epochs	50
	Threshold	0.2

model[3], which was trained using over 3 million words and phrases.

Kim et al. [3] presented the use of a non-static CNN instead of static CNN to further fine-tune the word2vec embedding during the training of the CNN model for text classification tasks. He found that non-static CNN performs better for most of the tasks that he experimented on. However, Toh et al. [5] and Khalil et al. [11] followed only the static approach for aspect extraction, where the word2vec embeddings for each word are kept fixed during the training time. Fine-tuning of word embedding features can be useful when using word2vec models that are trained using a corpus different from the dataset that is used to train the CNN model. Especially for aspect extraction, if both datasets are from different domains (restaurant reviews vs laptop domain) and generated using different sources (e.g. online articles vs customer reviews), then the syntactic-semantic patterns and vocabulary used may not be the same for both datasets. Therefore, we experimented with both static and non-static model variations [3] of our improved CNN to test our hypothesis.

Toh et al [5] used Adadelta [14] as the update function. We used Adam as the optimizer of both CNN models, which is shown to converge faster than most of the existing optimization techniques [15]. We used k-fold cross validation with {k=5} to determine the best neural network configuration and values for hyperparameters (except for h_1 and h_2). We set 100 as maximum word count (n) for any sentence. Table I shows the hyperparameters used with baseline CNN, which are similar to the parameters selected by Toh et al. [5]. Table II presents the hyperparameters of improved CNN that are tuned for both domains using the cross-validation results and the equations (1) and (2) that are used determine the number of hidden units for each hidden layer.

C. Support Vector Machine

We used features used in Jihan et al. [4] to create SVMs for aspect category classification. Multi-label classification required to classify the aspect terms is performed with one-vs-rest strategy, as the SVM classifier itself is a binary classifier. Therefore, following a one-vs-rest strategy we used 12 and 82 SVM classifiers for the restaurant and laptop domains, respectively. We used cross-validation for selecting the optimal parameters for the classifier. Following is the list of features we used in our research:

TABLE IIII
HYPER-PARAMETERS OF IMPROVED CNN

Layer	Parameter	Rest.	Laptop
Convolutional Layer	Window size	3, 5	3, 5
	# of filters	300 each	300 each
	Activation	tanh	tanh
Dropout Layer	Dropout rate	0.7	0.7
Hidden Layer 1	# of Neurons	191	467
	Activation	ReLU	ReLU
Hidden Layer 2	# of Neurons	143	445
	Activation	ReLU	ReLU
Training parameters	Batch size	50	50
	Epochs	60	60
	Threshold	0.2	0.2

1) *Bag of Words*: Text represented as the multiset of its lemmatized words
2) *Custom built word lists*: Count of words in a collection of food and drink names / laptop related keywords
3) *Frequent words*: Count of frequent words per category based on tf-idf score identified in training dataset.
4) *Opinion Targets*: Extracted opinion targets annotated in the training dataset. Count of words per required target is identified by opinion target.
5) *Symbols*: Presence of price indicators and presence of exclamation mark.
6) *Ending Words*: Bag of five words at the end of a sentence.
7) *Named Entities*: Presence of a person, organization, product or location in the text.
8) *Head Nouns*: Presence of a head nouns extracted by per sentence phrase.
9) *Mean Embedding*: Mean embedding vector for each sentence was calculated using the word2vec Google`s pre-trained model[4].

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

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

TABLE III
EXPERIMENTAL RESULTS FOR STATIC AND NON-STATIC CNN WITH EACH WORD2VEC MODEL

word2vec	Restaurant F1		Laptop F1	
	Static	Non-Static	Static	Non-Static
CBOW trained	0.6700	0.6849	0.4229	0.4422
Skip-gram trained	0.7405	0.7481	0.4694	0.4880
Google word2vec	**0.7538**	**0.7596**	**0.493**	**0.5174**

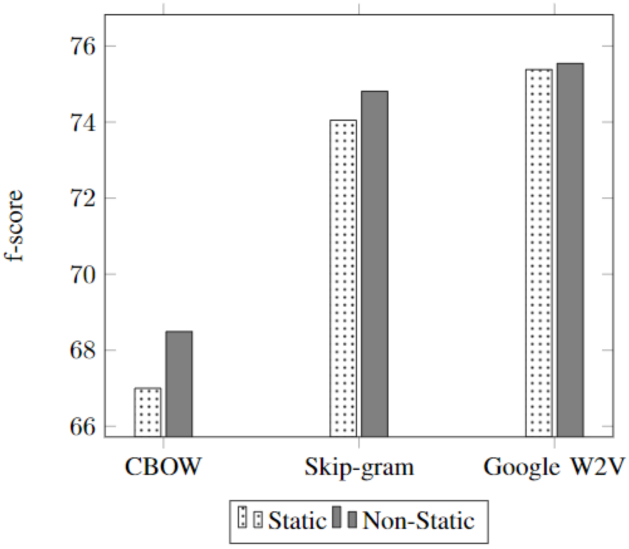

Fig. 2 F1-Score against word2Vec on Restaurant Dataset

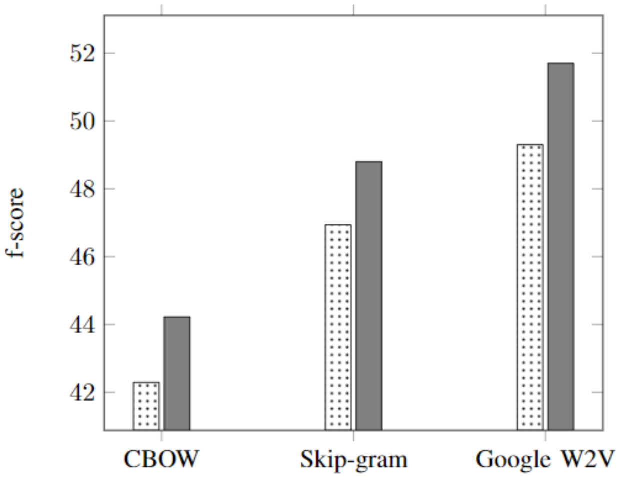

Fig. 3 F1-Score against Word2Vec on Laptop Dataset

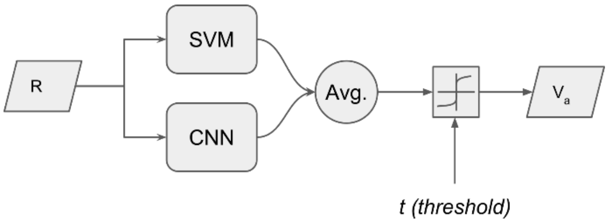

Fig. 4 Mixture of Classifiers for Aspect Classification (***R*** indicates the input review sentence; *SVM* and *CNN* refers to pretrained models discussed in previous sections; ***Avg.*** represents average function and $\boldsymbol{V_a}$ is the output aspect vector)

D. Mixture of Classifiers

First, CNN and SVMs are trained individually following the procedure explained in Section 4. Each model can estimate the probability of each aspect being presented in a given review. Thus, in the mixture of classifiers, we consider the probability outputs from both models to determine the class labels of each prediction. Let us consider $p_k(c)$ the probability of class $c \in C$, where k is either CNN classifier or one-vs-rest SVM classifiers. Therefore, the output probability of the mixture of classifiers $p_{mc}(c)$ is defined as illustrated in Equation 3. A visual illustration of the same is provided in Fig. **4**.

$$p_{mc}(c) = \frac{p_{cnn}(c)+p_{svm}(c)}{2}; \text{for } c \in C \quad (3)$$

In Equation 3, the final probability of each class is computed by averaging the probability output for each classifier. The resulting probability is then considered the prediction of the mixture of classifiers. Since the output is a probability value, we use a threshold to decide the actual classification; the predicted aspect labels. A suitable threshold is determined by using k-fold cross validation (similar settings to the hyperparameter tuning).

TABLE IV
RESULT COMPARISON WITH BASELINE AND BENCHMARK MODELS

Model	Restaurant F1	Laptop F1
CNN (baseline)	0.7356	0.4824
CNN (improved: L_1 only)	0.7492	0.5044
CNN (improved)	0.7596	0.5174
SVM [5]	0.7418	0.5221
NLANGP [8]	0.7303	0.5194
MTNA [7]	0.7642	-
Hybrid Model (t = 0.3)	**0.7717**	**0.5454**

V. DISCUSSION

Table III presents the F1 scores of the improved CNN model for both restaurant and laptop domains. Results are shown for each word2vec type used to initiate the word vectors to train the CNN models. Moreover, Table III shows the change of accuracy from static models to non-static models for each word2vec used. FigureFig. 2 andFig. 3 show the improvement of the models with different word2vec models for each static and non-static version with both Restaurant and Laptop datasets, respectively.

Using skip-gram trained word2vec, we were able to increase the accuracy of the CNN model significantly compared to the CBOW trained word2vec model. This is not surprising, as we have seen that Skip-gram models are significantly better on semantic tasks than CBOW models [8]. Aspect extraction also mostly involves understanding the semantic word relationships rather than interpreting the syntactic relationships between words.

However, the CNN model that used the pre-trained Google word2vec model gave better accuracy than when using other word2vec models that were trained using Yelp and Amazon review datasets. This is because those review datasets are much smaller (in the number of documents and vocabulary) than the Google news dataset that was used to train the pre-trained Google word2vec. Kim [3] shows that even though non-static CNN models are expected to perform better than static CNN models, it is not true for all the cases. However, aspect extraction for restaurant or laptop domain is a domain-specific task and it requires word vectors to be fine-tuned for that specific domain. Therefore, non-static CNN models performed better than static CNN models with the fine-tuned word vectors for the considered task and domains.

Table IV shows the best F1 scores for both baseline and improved CNN compared with the existing state of the art systems. CNN (baseline) and CNN (improved) are the baseline CNN and improved CNN, respectively. We also added the results of the improved CNN before optimizing the number of hidden layers and hidden units. Therefore, the CNN (improved: L1 only) uses a single hidden layer with 100 hidden neurons as similar to the baseline model.

The improved CNN has achieved a remarkable improvement compared to the baseline CNN model. This shows the significance of the modifications to the improved CNN model. If we compare CNN (baseline) and CNN (improved: L1 only), the modifications to the feature extraction and fine-tuning have shown a significant improvement of the CNN model. Moreover, optimizing the number of hidden layers and hidden units using the two hidden layer feed-forward network that was proposed by Huang et al. [7] has a noticeable contribution to the overall improvement of the CNN models for both restaurant and laptop domains.

Moreover, we can observe that improved CNN has a significant improvement for the restaurant domain. Our CNN model outperforms the hybrid system presented by Toh et al. [5] that combines both CNN and Feedforward Neural Network (FNN), and the one-vs-rest SVMs presented by Jihan et al. [4]. It is important to highlight that both above models use more features including word embeddings and they use strong classification models such as FNN and SVM. Yet, we showed that even by adding little flexibility to the CNN kernel with multiple kernels (e.g. CNN (improved: L1 only)), we can improve the feature selections to outperform the classification models that use both neural and traditional features. However, CNN alone has failed to outperform the MTNA. In contrast to the MTNA, our CNN architecture is simpler. Hence, instead of compromising the simplicity and computational complexity of the CNN architecture, we have outperformed the MTNA using our mixture of classifiers; which utilizes the both automatically extracted features and manually engineered features to extract the aspect from customer reviews.

Even though our CNN model shows close performance to the laptop domain results of both benchmark models [4, 5], it fails to outperform those models. We can explain this observation using the evaluation results of static and non-static variations of the CNN model. We can observe a significant improvement for the non-static model when compared with the static version for the Laptop domain, whereas for the

Restaurant review dataset that improvement is not that significant. Therefore, we can assume that the Google word2vec embeddings are semantically relevant to the restaurant domain, and less accurate for laptop domain. The significant improvement due to using non-static CNN opposed to static CNN for Laptop domain provides evidence to the poor accuracy of Google word2vec embedding for laptop domain. The fine-tuning of the non-static model increased the results remarkably from 0.4930 to 0.5174, which brings us closer to the benchmark models. Yet, this fine-tuning fails to improve the word embedding after a certain level (otherwise eventually we could have observed the same accuracy with every word2vec model used). The benchmark models used additional features specially designed for each domain, whereas we used only the Google pre-trained word2vec embeddings that are not optimized for laptop domain, which explains the failure of our CNN model to outperform benchmark models for the laptop domain.

Yet, our hybrid classifier has yielded a 4-5% accuracy gain compared to the state-of-the-art aspect extraction techniques in laptop domain. The CNN model illustrated comparably poor accuracy due to the insufficient domain specific evidence to the model. However, SVMs with manually engineered features have shown to capture such domain specific features remarkably [4]. Therefore, the use of SVMs probabilities to strength the Softmax outputs of CNN classifier has allowed us to incorporate that domain-specific evidence to strengthen the final probability outcomes of the hybrid model.

VI. Conclusion

This paper presents a mixture of classifiers for multi-domain aspect extraction, which can outperform the current state-of-the-art aspect extraction techniques by combining a CNN and one-vs-rest SVM classifiers.

First, we presented an improved CNN for aspect extraction, which can outperform the state-of-the-art systems when provided with well-trained word2Vec embeddings. Moreover, we showed that word embedding features generated using skip-gram trained models are better than the features from CBOW trained word2vec models for aspect extraction. We also demonstrated how the size and the domain of corpus used can affect the accuracy of CNN models used for aspect extraction. Our experiment shows that non-static CNN models can be used to improve aspect extraction in the absence of word2vec models trained with domain-specific corpora.

Moreover, we have improved the CNN model by introducing a second hidden layer. We have shown that using the equations proposed by Huang et al. [7] to determine the number of hidden units of both layers can outperform the traditional CNN models with a single dense layer. We are expecting to further explore the effect of this modification to the CNN model for general text classification tasks.

Secondly, we showed that our improved CNN model can achieve comparable performance for both restaurant and laptop domains, without any domain-specific hyperparameter optimizations. Our experiments highlight an important observation; that the same model can be used in different domains effectively with the same set of hyperparameters that is optimized for another domain. We are yet to determine the general applicability of this observation by experimenting with data sets from different domains. If the hyperparameter optimization of our improved CNN model proves to be domain independent, this will make the use of this CNN model on a new domain more straightforward, since no domain-specific parameter optimization is needed.

Finally, we derived a mixture of classifiers combining our improved CNN model with the SVM classifiers based on state-of-the-art custom engineered features, without introducing additional complexity to the improved CNN architecture. We demonstrated that the combined accuracy of CNN and SVM classifiers to outperform the current best systems for both restaurant and laptop domains.

In the future, we expect to extend the CNN architecture and to experiment with new deep neural architectures for aspect extraction from multi-domain customer reviews. The attention technique can be a possible direction to further improving deep neural networks for the task of aspect extraction. Moreover, exploring the new ways of building embeddings models; capturing both general and domain-specific data can enable new avenue of research for the both aspect extraction and text classification tasks.

References

[1] C. Lin and Y. He, “Joint sentiment/topic model for sentiment analysis,” in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009.

[2] K. Schouten and F. Frasincar, “Survey on aspect-level sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 28, pp. 813-830, 2016.

[3] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882,* 2014.

[4] N. Jihan, Y. Senarath, D. Tennekoon, M. Wickramarathne and S. Ranathunga, “Multi-Domain Aspect Extraction Using Support Vector Machines,” in *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017)*, Taipei, 2017.

[5] Z. Toh and J. Su, “NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features.,” in *SemEval@ NAACL-HLT*, 2016.

[6] B. Wang and M. Liu, *Deep Learning for Aspect-Based Sentiment Analysis,* Stanford University report, https://cs224d. stanford. edu/reports/WangBo. pdf, 2015.

[7] G.-B. Huang, “Learning Capability and Storage Capacity of Two-hidden-layer Feedforward Networks,” *Trans. Neur. Netw.*, vol. 14, pp. 274-281, 3 2003.

[8] T. Mikolov, K. Chen, G. S. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *CoRR,* vol. abs/1301.3781, 2013.

[9] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq and others, “SemEval-2016 task 5: Aspect based sentiment analysis,” in *ProWorkshop on Semantic Evaluation (SemEval-2016)*, 2016.

[10] J. Machácek, “BUTknot at SemEval-2016 Task 5: Supervised Machine Learning with Term Substitution Approach in Aspect Category Detection.,” in *SemEval@ NAACL-HLT*, 2016.

[11] T. Khalil and S. R. El-Beltagy, “NileTMRG at SemEval-2016 Task 5: Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction.,” in *SemEval@ NAACL-HLT*, 2016.

[12] W. Xue, W. Zhou, T. Li and Q. Wang, “MTNA: a neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews,” in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2017.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks

from Overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[14] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[16] N. Jihan, Y. Senarath and S. Ranathunga, "Aspect Extraction from Customer Reviews Using Convolutional Neural Networks," in *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2018.