

Parameterized Convolutional Neural Networks for Aspect Level Sentiment Classification

Binxuan Huang

School of Computer Science
Carnegie Mellon University
binxuanh@cs.cmu.edu

Kathleen M. Carley

School of Computer Science
Carnegie Mellon University
kathleen.carley@cs.cmu.edu

Abstract

We introduce a novel parameterized convolutional neural network for aspect level sentiment classification. Using parameterized filters and parameterized gates, we incorporate aspect information into convolutional neural networks (CNN). Experiments demonstrate that our parameterized filters and parameterized gates effectively capture the aspect-specific features, and our CNN-based models achieve excellent results on SemEval 2014 datasets.

1 Introduction

Continuous growing of user generated text in social media platforms such as Twitter drives sentiment classification increasingly popular. The goal of sentiment classification is to detect whether a piece of text expresses a positive, a negative, or a neutral sentiment (Rosenthal et al., 2017). The majority of the literature focuses on general sentiment analysis (document level), not involving a specific topic or entity. When there are multiple aspects about an entity in a sentence, it is hard to determine the sentiment as a whole.

Differing from general sentiment classification, aspect level sentiment classification identifies opinions from text about specific entities and their aspects (Pontiki et al., 2015). For example, given a sentence “great food but the service was dreadful”, the sentiment polarity about aspect “food” is positive while the sentiment polarity about “service” is negative. If we ignore the aspect information, it is hard to determine the sentiment for a target aspect, which accounts for a large portion of sentiment classification error (Jiang et al., 2011).

Recently, machine learning based approaches are becoming popular for this task. Representative approaches in literature include Support Vector Machine (SVM) with manually created features (Jiang et al., 2011; Wagner et al., 2014) and

neural network based models (Tang et al., 2016; Wang et al., 2016; Huang et al., 2018). Because of neural networks’ capacity of learning representations from data without feature engineering, they are of growing interest for this natural language processing task. The mainstream neural methods are either based on long short-term memory (Hochreiter and Schmidhuber, 1997) or memory networks (Sukhbaatar et al., 2015). None of them are using convolutional neural networks (CNN), which are good at capturing local patterns.

In the present work, we propose two simple yet effective convolutional neural networks with aspect information incorporated. The overall architecture differs significantly from previous work. Specifically, we design two novel neural units that take target aspects into account. One is parameterized filter, the other is parameterized gate. These units both are generated from aspect-specific features and are further applied on the sentence. Our experiments show that our two model variants work surprisingly well on this type of task.

2 Related Work

Aspect level sentiment classification is a branch of sentiment classification (Pang et al., 2002; Wang and Manning, 2012). It aims at identifying the sentiment polarity of one aspect target in a context sentence.

One typical early work tries to identify the aspect level sentiment polarity based on predefined language rules (Nasukawa and Yi, 2003). Nasukawa and Yi first perform dependency parsing on sentences. Then rules are applied to determine the sentiment about aspects. Standard machine learning algorithms like SVM are also widely used on this task. Jiang et al. create several target-dependent features, then they feed these target-dependent features with content features into an

SVM classifier.

In recent years, aspect level sentiment classification is dominated by neural network based approaches. The majority of published works rely on the architecture of long short-term memory (LSTM) (Tang et al., 2016). Wang et al. (2016) use an attention vector generated from aspect embedding to better capture the important parts in sentences. Tay et al. (2018) introduce a word-aspect fusion operation to learn associative relationships between aspects and sentences. Huang et al. (2018) use an attention-over-attention layer to further improve the performance.

Another type of neural architectures known as memory network (Sukhbaatar et al., 2015) has also been used in this task. Tang et al. (2016) takes an aspect term as a query sent to external memory. Their model consists of multiple computational layers. Each layer is an attention model. One recent work Dyadic MemNN (Tay et al., 2017) places associative layers on top of memory networks to improve the performance.

The overall architecture in this paper differs significantly with all these previous works. To the best of our knowledge, this paper is the first attempt using convolutional neural networks (Kim, 2014; Huang and Carley, 2017) for aspect level sentiment classification.

3 Parameterized Convolutional Neural Networks

In this section, we introduce our method for aspect level sentiment classification, which is based on convolutional neural networks. We first describe CNN for general sentiment classification, then we introduce our two model variants Parameterized Filters for Convolutional Neural Networks (PF-CNN) and Parameterized Gated Convolutional Neural Networks (PG-CNN).

3.1 Problem Definition

In aspect level sentiment classification, we are given a sentence $s = [w_1, w_2, \dots, w_i, \dots, w_n]$ and an aspect target $t = [w_i, w_{i+1}, \dots, w_{i+m-1}]$. The goal is to classify whether the sentiment towards the aspect in the sentence is positive, negative, or neutral.

3.2 Convolutional Neural Networks

We first briefly describe convolutional neural networks (CNN) for general sentiment classification

(Kim, 2014). Given a sentence $s = [w_1, w_2, \dots, w_i, \dots, w_n]$, let $v_i \in R^k$ be the word vector for word w_i . A sentence of length n can be represented as a matrix $\mathbf{s} = [v_1, v_2, \dots, v_n] \in R^{n \times k}$. A convolution filter $w \in R^{h \times k}$ with width h is applied to the word matrix to get high-level representative features. Specifically, for a word window $v_{i:i+h-1} \in R^{h \times k}$, a feature c_i is generated by

$$c_i = f(w \odot v_{i:i+h-1} + b) \quad (1)$$

where \odot represents element-wise product, $b \in R$ is a bias term and f is a non-linear function. Sliding the filter window from the beginning of the word matrix till the end, we get a feature map $\mathbf{c} \in R^{n-h+1}$.

$$\mathbf{c} = [c_1, c_2, \dots, c_{n-h+1}] \quad (2)$$

After that, a pooling operation is applied over the feature map to get one single general sentiment feature θ_g in each map. We use max pooling in the CNN for sentences.

$$\theta_g = \text{pooling}(\mathbf{c}) \quad (3)$$

We denote this process as $\theta_g = \text{CNN}_g(\mathbf{s}; w, b)$. Using d such convolutional filters, we can get a general sentiment feature vector $\Theta_g \in R^d$ without information from aspect terms.

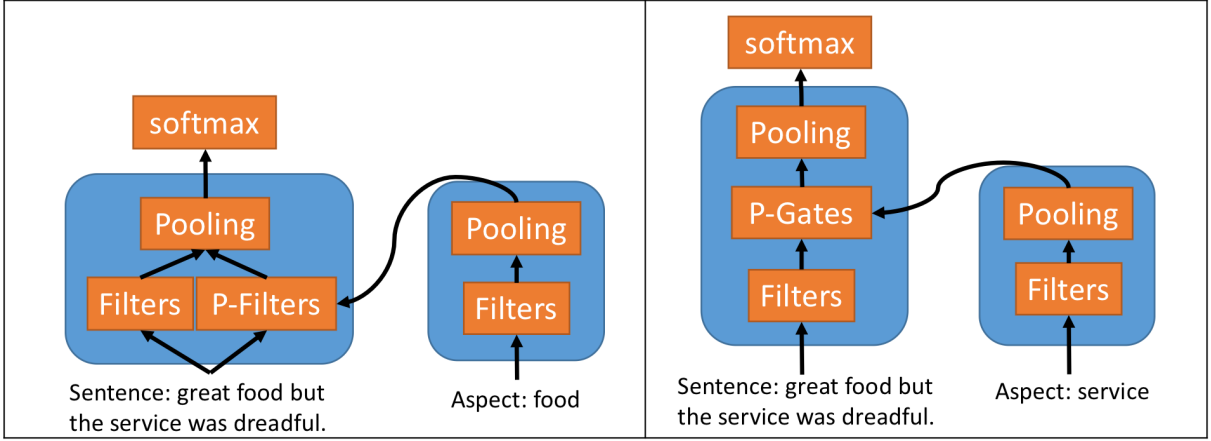
3.3 Parameterized Filters

Standard convolutional neural networks do not consider information from aspect terms. Herein, our first model variant overcomes this issue by parameterizing filters using aspect terms. We call it Parameterized Filters for Convolutional Neural Networks (PF-CNN). The overall architecture is shown in the left of Figure 1.

Formally, given the aspect term with length m , $t = [w_i, w_{i+1}, \dots, w_{i+m-1}]$ and the corresponding embedding matrix $\mathbf{t} = [v_i, v_{i+1}, \dots, v_{i+m-1}] \in R^{m \times k}$, we first use another CNN_t to extract one single feature θ_t from \mathbf{t} .

$$\theta_t = \text{CNN}_t(\mathbf{t}; w_t, b_t) \quad (4)$$

where $w_t \in R^{h_t \times k}$, b_t are the convolutional filter, bias term for CNN_t . h_t is the width of filters applied on aspect targets. With $h_s \times k$ such filters and bias terms, we can get a feature matrix $\Theta_t \in R^{h_s \times k}$, where h_s is the filter width applied on sentences. We use average pooling in the CNN_t for aspects.



PF-CNN

PG-CNN

Figure 1: The overall architectures of PF-CNN and PG-CNN.

In the next step, Θ_t is further used as a convolutional filter applied on the sentence.

$$\theta_s = CNN_s(s; \Theta_t, b_s) \quad (5)$$

Using such d parameterized filters, we get the aspect-specific features $\Theta_s \in R^d$ with target term information. We further concatenate the targeted feature vector with general sentiment features as the final classification features $\Theta = [\Theta_g, \Theta_s]$.

3.4 Parameterized Gates

The second model variant we designed is called Parameterized Gated Convolutional Neural Networks (PG-CNN). The overall architecture is shown in the right of Figure 1.

Similar with PF-CNN, PG-CNN also utilizes a convolutional neural network to extract feature Θ_t from aspect terms, which instead is used as a gate (Dauphin et al., 2017) in the CNN applied on the sentence. The key step of PG-CNN is described in equation (6).

$$c_i = (w \odot v_{i:i+h-1} + b) \cdot \sigma(\Theta_t \odot v_{i:i+h-1} + b) \quad (6)$$

Instead of using a non-linear function f in equation (1), we use a gate $\sigma(\Theta_t \odot v_{i:i+h-1} + b)$ to control how much information passing to the next layer, where $\sigma(\cdot)$ is sigmoid function. For each general filter applied on the sentence, one parameterized gate is generated from the aspect.

After that, we generate the final classification feature Θ in the same way as standard CNN.

3.5 Final Classification

We feed the final classification feature into a linear layer to project Θ into the space of targeted classes:

$$x = W_l \cdot \Theta + b_l \quad (7)$$

where W_l and b_l are the weight matrix and bias. Following the linear layer, we use a softmax layer to compute the probability of class c .

$$P(y = c|x) = \frac{\exp(x_c)}{\sum_{i \in C} \exp(x_i)} \quad (8)$$

3.6 Model training

We train our model to minimize the cross-entropy loss function with L_2 regularization:

$$loss = - \sum_{(s,t)} \sum_{c \in C} I(y = c) \log P(y = c|s, a) + \lambda \|p\|^2$$

where $I(\cdot)$ is the indicator function and p is the set of all parameters in the convolutional layers and linear layer.

4 Experiments

4.1 Experiments Setting

Dataset

We adopt one widely used dataset from SemEval 2014 Task 4 (Pontiki et al., 2014). It contains two domain-specific datasets for laptops and restaurants. Each data point is a pair of a sentence and an aspect term. Experienced annotators tagged each pair with sentiment polarity. Following recent work (Tay et al., 2018), we take 500 training instances as development set¹. Unfortunately, many works have not mentioned the usage of development set (Wang et al., 2016; Ma et al., 2017).

Hyperparameters and Training

We use rectifier as non-linear function f in the CNN_g , CNN_t and sigmoid in the CNN_s , filter window sizes of 1, 2, 3, 4 with 100 feature maps

¹The splits can be found at https://github.com/vanzytay/ABSA_DevSplits.

Dataset	Positive	Neutral	Negative
Laptop-Train	767	373	673
Laptop-Dev	220	87	193
Laptop-Test	341	169	128
Restaurant-Train	1886	531	685
Restaurant-Dev	278	102	120
Restaurant-Test	728	196	196

Table 1: Statistics of the datasets.

Model	Laptops		Restaurants	
	3-way	Binary	3-way	Binary
TD-LSTM	62.38	79.31	69.73	84.41
AT-LSTM	65.83	78.25	74.37	84.74
ATAE-LSTM	60.34	74.20	70.71	84.52
AF-LSTM	68.81	83.58	75.44	87.78
CNN	68.65	85.50	77.95	89.50
PF-CNN	70.06	86.35	79.20	90.15
PG-CNN	69.12	86.14	78.93	90.58

Table 2: Comparisons results with baselines. We use accuracy to measure the performance. Performances of baselines are cited from (Tay et al., 2018).

each, l_2 regularization term of 0.001 and mini-batch size of 25. Parameterized filters and gates have the same size and number as normal filters. They are generated uniformly by CNN with window sizes of 1, 2, 3, 4, eg. among 100 parameterized filters with size 3, 25 of them are generated by aspect CNN with filter size 1, 2, 3, 4 respectively. The word embeddings are initialized with 300-dimensional Glove vectors (Pennington et al., 2014) and are fixed during training. For the out of vocabulary words we initialize them randomly from uniform distribution $U(-0.01, 0.01)$. We apply dropout on the final classification features of PG-CNN. The dropout rate is chosen as 0.3.

Training is done through mini-batch stochastic gradient descent with Adam update rule (Kingma and Ba, 2015). The initial learning rate is 0.001. If the training loss does not drop after every three epochs, we decrease the learning rate by half. We adopt early stopping based on the validation loss on development sets.

4.2 Results

We use accuracy metric to measure the performance. To show the effectiveness of our model, we compare it with several baseline methods. We list them as follows:

TD-LSTM uses two LSTM networks to model the preceding and following contexts surrounding

the aspect term. The last hidden states of these two LSTM networks are concatenated for predicting the sentiment polarity (Tang et al., 2016).

AT-LSTM combines the sentence hidden states from a LSTM with the aspect term embedding to generate the attention vector. The final sentence representation is the weighted sum of the hidden states (Wang et al., 2016).

ATAE-LSTM further extends AT-LSTM by appending the aspect embedding into each word vector (Wang et al., 2016).

AF-LSTM introduces a word-aspect fusion attention to learn associative relationships between aspect and context words (Tay et al., 2018).

CNN uses the architecture proposed in (Kim, 2014) without explicitly considering aspect. We use filter window sizes of 1,2,3,4 with 100 maps each. Dropout rate is chosen as 0.5. Early stopping based on validation accuracy is applied.

Our two models achieve the best performance when compared to these baselines as shown in Table 2, which shows that our proposed neural units effectively captures the aspect-specific features. Compared to one recently proposed model AF-LSTM, our method achieve 2%-5% improvements. Surprisingly, a vanilla CNN works quite well on this problem. It even beats these well-designed LSTM models, which further proves that using CNN-based methods is a direction worth exploring.

4.3 Case Study & Discussion

Compared to a vanilla CNN, our two model variants could successfully distinguish the describing words for corresponding aspect targets. In the sentence “the appetizers are ok, but the service is slow”, a vanilla CNN outputs the same negative sentiment label for both aspect terms “appetizers” and “service”, while PF-CNN and PG-CNN successfully recognize that “slow” is only used for describing “service” and output neutral and negative labels for aspects “appetizers” and “service” respectively. In another example “the staff members are extremely friendly and even replaced my drink once when i dropped it outside”, our models also find out that positive and neutral sentiment for “staff” and “drink” respectively.

One thing we notice in our experiment is that a vanilla CNN ignoring aspects has comparable performance with some well-designed LSTM models in this aspect-level sentiment classification task.

For a sentence containing multiple aspects, we assume the majority of the aspect-level sentiment label is the sentence-level sentiment label. Using this labeling scheme, in the restaurant data, 1034 out of 1117 test points have the same sentence-level and aspect-level labels. Thus, a sentence-level classifier with accuracy 75% also classifies 70% aspect-labels correctly. A similar observation was made for the laptop dataset as well. Probably this is the reason why a vanilla CNN has comparable performance on these two datasets. For future research, a more balanced dataset would be helpful to overcome this issue.

5 Conclusion

We propose a novel method for aspect level sentiment classification. We introduce two novel neural units called parameterized filter and parameterized gate to incorporate aspect information into the convolutional neural network architecture. Comparisons with baseline methods show our model effectively learns the aspect-specific sentiment expressions. Our experiments demonstrate a significant improvement compared to multiple strong neural baselines.

To the best of our knowledge, our model is the first attempt using convolutional neural networks solving this problem. We hope this work could inspire future research exploring in this direction. It is also interesting to see whether such parameterized CNN architecture could benefit other natural language processing tasks involving text pairs like question answering task.

Acknowledgments

We would like to thank the reviewers for their helpful comments that greatly improved the article. We would also like to thank Sumeet Kumar for his valuable suggestions.

References

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th international conference on machine learning*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Binxuan Huang and Kathleen M Carley. 2017. On predicting geolocation of tweets using convolutional

neural networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 281–291. Springer.

- Binxuan Huang, Yanglan Ou, and Kathleen M. Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *Social, Cultural, and Behavioral Modeling*, pages 197–206. Cham. Springer International Publishing.
- Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4068–4074.
- Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- 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)*, pages 1532–1543.
- 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)*, pages 486–495.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and

- Suresh Manandhar. 2014. *Semeval-2014 task 4: Aspect based sentiment analysis*. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.
- Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307.
- Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.
- Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *AAAI*.
- Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. Dyadic memory networks for aspect-based sentiment analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 107–116. ACM.
- Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4.
- Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.
- Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, pages 606–615.