

# Syntax-Aware Aspect-Level Sentiment Classification with Proximity-Weighted Convolution Network

Chen Zhang  
Beijing Institute of Technology  
Beijing, China  
& Zhejiang Lab  
Hangzhou, China  
gene@bit.edu.cn

Qiuchi Li  
University of Padua  
Padua, Italy  
qiuchili@dei.unipd.it

Dawei Song\*  
Beijing Institute of Technology  
Beijing, China  
dwsong@bit.edu.cn

## ABSTRACT

It has been widely accepted that Long Short-Term Memory (LSTM) network, coupled with attention mechanism and memory module, is useful for aspect-level sentiment classification. However, existing approaches largely rely on the modelling of semantic relatedness of an aspect with its context words, while to some extent ignore their syntactic dependencies within sentences. Consequently, this may lead to an undesirable result that the aspect attends on contextual words that are descriptive of other aspects. In this paper, we propose a proximity-weighted convolution network to offer an aspect-specific syntax-aware representation of contexts. In particular, two ways of determining proximity weight are explored, namely position proximity and dependency proximity. The representation is primarily abstracted by a bidirectional LSTM architecture and further enhanced by a proximity-weighted convolution. Experiments conducted on the SemEval 2014 benchmark demonstrate the effectiveness of our proposed approach compared with a range of state-of-the-art models<sup>1</sup>.

## KEYWORDS

Sentiment classification, Syntax-awareness, Proximity-weighted convolution

### ACM Reference Format:

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Syntax-Aware Aspect-Level Sentiment Classification with Proximity-Weighted Convolution Network. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*, July 21–25, 2019, Paris, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3331184.3331351>

## 1 INTRODUCTION

Aspect-level sentiment classification, also called aspect-based sentiment classification, is a fine-grained sentiment classification task aiming at identifying the polarity of a given aspect within a certain

context, i.e. a comment or a review. For example, in the following comment about food “*They use fancy ingredients, but even fancy ingredients don’t make for good pizza unless someone knows how to get the crust right.*”, the sentiment polarities for aspects *ingredients*, *pizza* and *crust* are *positive*, *negative* and *neutral* respectively.

Aspect-level sentiment classification has attracted an increasing attention in the fields of Natural Language Processing (NLP) and Information Retrieval (IR), and plays an important role in various applications such as personalized recommendation. Earlier works in this area focused on manually extracting refined features and feeding them into classifiers like Support Vector Machine (SVM) [7], which is labor intensive. In order to tackle the problem, automatic feature extraction has been investigated. For example, Dong et al. [4] proposed to adaptively propagate the sentiments of context words to the aspect via their syntactic relationships. Vo and Zhang [15] built a syntax-free feature extractor to identify a rich source of relevant features. Despite the effectiveness of these approaches, Tang et al. [13] claimed that the modelling of semantic relatedness of an aspect and its context remained a challenge, and proposed to use target-dependent LSTM network to address this challenge.

As the attention mechanism and memory network have yielded good results in many NLP tasks such as machine translation [1, 9], LSTM combined with attention [6, 10] or memory network [3, 14] is deployed to aspect-level sentiment classification to aggregate contextual features for prediction. Being capable of modelling semantic interactions between aspects and their corresponding contexts, these models have improved performance over previous approaches. However, they generally ignore the syntactic relations between the aspect and its context words, which may hinder the effectiveness of aspect-based context representation. For instance, a given aspect may attend on several context words that are descriptively near to the aspect but not correlated to the aspect syntactically. As a concrete example, in “*Its size is ideal and the weight is acceptable.*”, the aspect term *size* may easily be depicted by *acceptable* based on the semantic relatedness, which is in fact not the case. Syntactic parsing has been used in some previous work [4], however, the word-level parsing could impede feature extraction across different phrases, as the sentiment polarity of an aspect is usually determined by a key phrase instead of a single word [5].

In order to address the limitations mentioned above, we propose an aspect-level sentiment classification framework that leverages the syntactic relations between an aspect and its context and aggregates features at the n-gram level, within a LSTM-based architecture. Inspired by the position mechanism [3, 6, 8, 14], the framework utilizes a context word’s syntactic proximity to the aspect, a.k.a

\*Corresponding author.

<sup>1</sup>Code is available at <https://github.com/GeneZC/PWCN>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '19, July 21–25, 2019, Paris, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6172-9/19/07...\$15.00

<https://doi.org/10.1145/3331184.3331351>

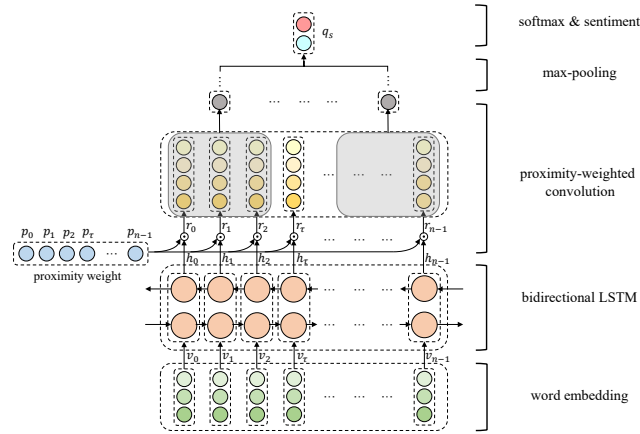


Figure 1: Overview of the model architecture.

proximity weight, to determine its importance in the sentence. We then integrate the proximity weights into a convolution network to capture n-gram information, called as **Proximity-Weighted Convolution Network (PWCN)**. Finally, a layer of max-pooling is adopted to select the most significant features for prediction.

Experiments are conducted on SemEval 2014 Task4 datasets. The results show that our model achieves a higher performance than a range of state-of-the-art models, and hence illustrate that syntactical dependencies are more beneficial than semantic relatedness to aspect-level sentiment classification.

## 2 THE PROPOSED MODEL

An overview of our proposed model is given in Figure 1. In the model, an  $n$ -word sentence containing a target  $m$ -word aspect term is formulated as  $S = \{w_0, w_1, \dots, w_\tau, w_{\tau+1}, \dots, w_{\tau+m-1}, \dots, w_{n-1}\}$ , where  $\tau$  denotes the start token of the aspect term. Each word is embedded into a low-dimensional real-valued vector with a matrix  $E \in \mathbb{R}^{|V| \times d_e}$  [2], where  $|V|$  is the size of dictionary while  $d_e$  is the dimensionality of a word vector. After word vectors  $V = \{e_0, e_1, \dots, e_\tau, e_{\tau+1}, \dots, e_{\tau+m-1}, \dots, e_{n-1}\}$  are obtained through word embedding, a bidirectional LSTM is adopted to produce the hidden state vectors  $H = \{h_0, h_1, \dots, h_\tau, h_{\tau+1}, \dots, h_{\tau+m-1}, \dots, h_{n-1}\}$ . Particularly,  $h_i \in \mathbb{R}^{2d_h}$  is a concatenation of hidden states respectively obtained from the forward LSTM and the backward LSTM, where  $d_h$  is the dimensionality of a hidden state vector in an unidirectional LSTM. The hidden state representation is further enhanced by proximity-weighted convolution and then used for prediction of sentiment polarity.

### 2.1 Proximity Weight

Previous attention-based models mainly focus on how to obtain a context representation based on its component words' semantic correlations with a corresponding aspect [3, 5, 6, 8, 10, 14]. These models calculate attention weights referring to word vector representation in the latent semantic space, without taking into consideration syntax information. This may limit the effectiveness of these models in term of mis-identify crucial context words for characterizing the aspect. Therefore, we replace this complicated modelling

of aspects by incorporating syntactical dependencies to uncover component words' characteristics to the aspect<sup>2</sup>. Such syntactical dependency information in our proposed model is formalized as *proximity weight*, which describes the contextual words' proximity to the aspect. Recall the example related to *weight* of a laptop saying that “*Its size is ideal and the weight is acceptable.*”. The set of words including  $\{ideal, acceptable\}$  that are closer to the aspect term *weight* in terms of semantics, should have a larger probability describing the weight of a laptop. Further, from the perspective of syntax parsing, *ideal* could be safely excluded from the word set as it is syntactically too far from *weight*. Actually, *acceptable* is the true descriptor of *weight*, indicating a positive sentiment.

Following this idea, we propose two different methods, namely *position proximity* and *dependency proximity*, to model the syntactical dependency between contextual words and the aspect term respectively.

**2.1.1 Position Proximity.** Generally, it is more likely to see that words around an aspect are describing the aspect. Thus, we view such position information as an approximated syntactical proximity measurement. Position proximity weights are computed by the formula below:

$$p_i = \begin{cases} 1 - \frac{\tau-i}{n} & 0 \leq i < \tau \\ 0 & \tau \leq i < \tau + m \\ 1 - \frac{i-\tau-m+1}{n} & \tau + m \leq i < n \end{cases} \quad (1)$$

where proximity weight  $p_i \in \mathbb{R}$ . Intuitively, the weight decreases in proportion to the word's distance to the nearest border of the aspect term.

**2.1.2 Dependency Proximity.** Apart from the absolute position in the context, we also consider measuring the distances between words in a syntax dependency parsing tree. For example, in a comment “*the food is awesome - definitely try the striped bass.*” with *food* as the aspect, we first construct a dependency tree<sup>3</sup>, then compute for a context word the tree-based distance, i.e. the length of the shortest path in the tree, between the word and *food*. If the aspect otherwise contains more than one word, we take the minimum of the tree-based distances between a context word and all the aspect component words. In the uncommon case where more than one dependency trees are present in a context, we manually set the distance between the aspect term and context words in other trees to a constant, i.e. half of the sentence length<sup>4</sup>.

For a better illustration of the proposed method, an example sentence is shown in Figure 2. With the above described approach, the sequence of tree-based distances for all words in the sentence with respect to the aspect term *aluminum*,  $\mathbf{d} = \{d_0, d_1, \dots, d_\tau, d_{\tau+1}, \dots, d_{\tau+m-1}, \dots, d_{n-1}\}$  are marked below the words in the figure. The dependency proximity weights of the sentence are then assigned as:

$$p_i = \begin{cases} 1 - \frac{d_i}{n} & 0 \leq i < \tau \quad \text{or} \quad \tau + m \leq i < n \\ 0 & \tau \leq i < \tau + m \end{cases} \quad (2)$$

<sup>2</sup>We have conducted experiments using proximity weight combined with attention weight, i.e. the combination of semantic relatedness and syntactic proximity, but we get unexpected sub-optimal results, which will be shown in experiments section.

<sup>3</sup>With spaCy toolkit: <https://spacy.io/>.

<sup>4</sup>It's a proper number that could serve as the boundary of possible descriptive contextual words in our experiments.

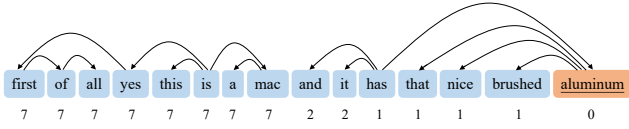


Figure 2: Dependency distance with respect to *aluminum*.

## 2.2 Proximity-Weighted Convolution

Compared with the use of word-level features, Aspect-level sentiment classification with phrase-level features have been shown more effective [5, 8]. We are thus inspired to propose a *proximity-weighted convolution*, which is essentially 1-dimensional convolution with a length- $l$  kernel, i.e.  $l$ -gram. Different from the original definition of convolution, the proximity-weighted convolution assigns proximity weight before convolution calculation. The proximity weight assigning process is formulated below:

$$r_i = p_i h_i \quad (3)$$

where  $r_i \in \mathbb{R}^{2d_h}$  represents the proximity-weighted representation of the  $i$ -th word in the sentence.

Additionally, we zero-pad the sentence to ensure the convolution outputs a sequence of the same length as the input sentence. The convolution process contains:

$$t = \left\lfloor \frac{l}{2} \right\rfloor \quad (4)$$

$$q_i = \max(\mathbf{W}_c^T [r_{i-t} \oplus \dots \oplus r_i \oplus \dots \oplus r_{i+t}] + b_c, 0) \quad (5)$$

where  $q_i \in \mathbb{R}^{2d_h}$  denotes the features extracted by convolution layer, and  $\mathbf{W}_c \in \mathbb{R}^{l \cdot 2d_h \times 2d_h}$  and  $b_c \in \mathbb{R}^{2d_h}$  are weight and bias of the convolution kernel, respectively.

As only few output features of the convolution layer are expected to be instructive for classification, we choose the most prominent feature  $q_s \in \mathbb{R}^{2d_h}$  through a 1-dimension max-pooling layer with a kernel of length  $n$ , such that:

$$q_s = [\max_{0 \leq i < n} q_{i,j}]^T \quad 0 \leq j < 2d_h \quad (6)$$

where  $q_{i,j}$  is the  $j$ -th element of  $q_i$ .

Finally, the most prominent feature vector  $q_s$  is fed to a fully connected layer, followed by a softmax normalization to obtain the distribution  $y \in \mathbb{R}^{d_p}$  over the decision space on  $d_p$ -way sentiment polarity:

$$y = \text{softmax}(\mathbf{W}_f^T q_s + b_f) \quad (7)$$

where  $b_f \in \mathbb{R}^{d_p}$  is the bias of the fully connected layer and  $\mathbf{W}_f \in \mathbb{R}^{2d_h \times d_p}$  is the learned weight.

Our model is trained by the standard gradient descent algorithm, with the loss being the cross entropy loss with  $L_2$  regularization:

$$L = - \sum_{(s, \hat{y}) \in \mathbf{D}} \sum_u \hat{y}_u \log y_u + \lambda \|\Theta\|_2 \quad (8)$$

Here,  $\hat{y}$  means one-hot vector of golden label while  $\mathbf{D}$  is the collection of (sentence, label) pairs. And  $\Theta$  denotes all trainable parameters,  $\lambda$  is the coefficient of  $L_2$  regularization.

## 3 EXPERIMENTS

### 3.1 Datasets and Experimental Settings

We conduct experiments on two benchmarking datasets from SemEval 2014 [12]. The datasets consist of reviews and comments from two categories: laptop and restaurant, respectively.

In all of our experiments, 300-dimensional GloVe is leveraged to initialize word embedding [11]. All parameters of our model are initialized with the uniform distribution. The dimensionality of hidden state vectors is set to 300. We use Adam as the optimizer with a learning rate of 0.001. The coefficient of  $L_2$  regularization is  $10^{-5}$  and batch size is 64. We adopt Accuracy and Macro-Averaged F1 as the evaluation metrics. Additionally, the length of  $n$ -gram is set to 3<sup>5</sup>.

### 3.2 Model Comparison

A comprehensive comparison is carried out between our proposed models, i.e. PWCN with position proximity (**PWCN-Pos**) and with dependency proximity (**PWCN-Dep**), against several state-of-the-art baseline models, as listed below:

- **LSTM** [13] only uses the last hidden state vector to predict sentiment polarity.
- **RAM** [14] considers hidden state vectors of context as external memory and applies Gated Recurrent Unit (GRU) structure to multi-hop attention. The top-most representation is used for predicting polarity.
- **IAN** [10] models attention between aspect and its context interactively with two LSTMs.
- **TNet-LF** [8] leverages Context-Preserving Transformation to preserve and strengthen the informative part of context. It also benefits from a multi-layer architecture.

We also present comparison with two variants of **PWCN-Pos**. Firstly, we propose **Att-PWCN-Pos** model, in which the proximity weight is multiplied by the normalized attention weight, to check whether semantic relatedness and syntax relationship could be incorporated with each other. Further, we intend to measure the effectiveness of  $n$ -gram via setting  $l$ -gram to 1-gram, which naturally degrades convolution process to point-wise feed-forward network, and we call it **Point-PWCN-Pos**.

### 3.3 Experimental Results

The experimental results in Table 1 are yielded by averaging the performances of 3 runs with random initialization. The results demonstrate the general effectiveness of PWCN, which largely outperforms LSTM, RAM and IAN, and also achieves some increase over TNet-LF, the best-performing baseline model under comparison. Among the two types of underlying syntactic structure of sentences captured by PWCN model, dependency proximity brings more benefits to the overall performance than position proximity, with consistently higher Macro-F1 scores on both datasets. The results also support our claim that  $n$ -gram information is critical for feature extraction, which can be observed from the disparity between Point-PWCN-Pos and PWCN-Pos.

Moreover, it is interesting to see that PWCN-based methods with solely syntactic information outperform the Att-PWCN model that

<sup>5</sup>We have tried several numbers and 3 performed the best.

**Table 1: Experimental results. Average accuracy and macro-F1 score over 3 runs with random initialization. The best results are in bold. The marker  $\dagger$  refers to  $p$ -value  $< 0.05$  when comparing with IAN, while the marker  $\ddagger$  refers to  $p$ -value  $< 0.05$  when comparing with TNet-LF. The relative increase over the LSTM baseline is given in bracket.**

Model	Laptop		Restaurant	
	Acc	Macro-F1	Acc	Macro-F1
LSTM	69.63	63.51	77.99	66.91
RAM	72.81 (+4.57%)	68.59 (+8.00%)	79.89 (+2.44%)	69.49 (+3.86%)
IAN	71.63 (+2.87%)	65.94 (+3.83%)	78.59 (+0.77%)	68.41 (+2.24%)
TNet-LF	75.16 (+7.94%)	71.10 (+11.95%)	80.20 (+2.83%)	70.78 (+5.78%)
Att-PWCN-Pos	72.92 (+4.72%)	68.14 (+7.29%)	80.15 (+2.77%)	70.17 (+4.87%)
Point-PWCN-Pos	74.45 (+6.92%)	69.47 (+9.38%)	80.00 (+2.58%)	69.93 (+4.51%)
PWCN-Pos	75.23 $\dagger$ (+8.17%)	70.71 $\dagger$ (+11.34%)	<b>81.12<math>\dagger\ddagger</math></b> (+4.01%)	71.81 $\dagger$ (+7.32%)
PWCN-Dep	<b>76.12<math>\dagger\ddagger</math></b> (+9.32%)	<b>72.12<math>\dagger\ddagger</math></b> (+13.56%)	80.96 $\dagger$ (+3.81%)	<b>72.21<math>\dagger</math></b> (+7.92%)

**Table 2: Visualization of a case with respect to *food***

Method	Visualization	Pred.
Att.	great food but the service was dreadful !	negative
Pos.	great food but the service was dreadful !	positive
Dep.	great food but the service was dreadful !	positive

combines syntactic and semantic information. While this shows the superiority of leveraging syntactical dependency information to using semantic relatedness, we further conjecture that the attention mechanism could erroneously render term dependencies thus adversely affect the correct decisions of PWCN.

### 3.4 Impact of Syntax

To understand the effect proximity weight has brought, we conduct a case study on an example which could be seen in Table 2. More specifically, we visualize the weights given by attention in Att-PWCN-Pos, position proximity in PWCN-Pos, and dependency proximity in PWCN-Dep separately along with their predictions.

We can observe that the existing attention mechanism makes wrong decision on which context word depicts *food* in an extreme way, while both sorts of proximity weight in our model handle this problem properly, which is within our expectation.

## 4 CONCLUSIONS AND FUTURE WORK

Previous methods of utilizing aspect information for the aspect-level sentiment classification depend on the modelling of aspect representation from a semantic perspective, while the syntactic relationship between the aspect and its context is generally neglected. In this paper, we have built a framework that leverages n-gram information and syntactic dependency between aspect and contextual terms into an applicable model. Experimental results have demonstrated the effectiveness of our proposed models and suggested that syntactic dependency is more beneficial to aspect-level sentiment classification than semantic relatedness.

We believe it is a promising direction to dive into concrete examples to analyze the difference between PWCN models and attention-based models to achieve a deep understanding of where the syntactical dependencies overwhelm semantic relatedness.

## ACKNOWLEDGEMENTS

This work is supported by The National Key Research and Development Program of China (grant No. 2018YFC0831700), Natural Science Foundation of China (grant No. U1636203), and Major Project Program of Zhejiang Lab (grant No. 2019DH0ZX01).

## REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.
- [2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.
- [3] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *EMNLP*. 452–461.
- [4] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In *ACL*. 49–54.
- [5] Chuang Fan, Qinghong Gao, Jiachen Du, Lin Gui, Ruifeng Xu, and Kam-Fai Wong. 2018. Convolution-based Memory Network for Aspect-based Sentiment Analysis. In *SIGIR*. New York, NY, USA, 1161–1164.
- [6] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In *COLING*. 1121–1131.
- [7] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *ACL*. 151–160.
- [8] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation Networks for Target-Oriented Sentiment Classification. In *ACL*. 946–956.
- [9] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*. 1412–1421.
- [10] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive Attention Networks for Aspect-level Sentiment Classification. In *IJCAI*. 4068–4074.
- [11] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.
- [12] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *SemEval*. 27–35.
- [13] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In *COLING*. 3298–3307.
- [14] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *EMNLP*. 214–224.
- [15] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent Twitter Sentiment Classification with Rich Automatic Features. In *IJCAI*. 1347–1353.