Research Article

Syntactic Knowledge Embedding Network for Aspect-Based Sentiment Classification

Yan Wen,1 Wenkai Li,1 Qingtian Zeng,1,2 Hua Duan,1 Feng Zhang,1 and Shitao Kang1

1College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
2Shandong Key Laboratory of Wisdom Mine Information Technology, Qingdao 266590, China

Correspondence should be addressed to Yan Wen; wenyannya@163.com and Qingtian Zeng; qtzeng@163.com

Received 4 January 2022; Accepted 23 March 2022; Published 15 April 2022

Copyright © 2022 Yan Wen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An aspect-based sentiment classification task is a fine-grained sentiment analysis task, which is aimed at identifying the sentiment polarity of a given aspect in subjective sentences. In recent years, some researchers have applied pretrained BERT models to this task. However, existing research only uses the BERT output layer and ignores the syntactic features in the middle layers, leading to deviations in the prediction results. In order to solve above problems, we propose a new model BERT-SFE. Firstly, we explicitly utilize the middle layers of BERT to capture the underlying syntactic features. Secondly, we construct a syntactic feature extraction unit based on Star-Transformer, which uses an auxiliary vector and the star network structure to capture both local and global syntactic information in a sentence. Finally, we merge the syntactic features with the semantic features from the BERT output layer in the feature fusion layer, obtaining a more accurate sentiment representation of the aspect. The experimental results on three public ABSA datasets show that using the syntactic feature extraction unit based on Star-Transformer to mine the syntactic knowledge in the middle layers of BERT can effectively improve the accuracy of sentiment classification. BERT-SFE achieves the best performance compared with existing models.

1. Introduction

Sentiment analysis is a popular task, which is aimed at identifying the user’s sentiment polarity from subjective texts containing sentimental information, is the computational study of people’s opinions, attitudes, and emotions toward an entity. Researchers use machine learning and deep learning approaches to capture complex dependencies within data [1–3]. There are three categories in the sentiment analysis task: document-based, sentence-based, and aspect-based sentiment analysis (ABSA) [4]. ABSA is a fine-grained sentiment analysis task, which mainly includes three subtasks: aspect-based sentiment classification [5, 6], aspect extraction, and sentiment word extraction [7]. Aspect-based sentiment classification is aimed at identifying the sentiment polarity of a given aspect in a subjective sentence. For example, given a comment sentence “Waiters are very friendly and the pasta is simply average,” the sentiment polarities for the aspects “Waiters” and “pasta” are “positive” and “negative”, respectively.

In recent years, scholars have conducted a lot of researches on the aspect-based sentiment classification task, which is modelled as a classification problem in most cases. This problem is commonly solved by sentiment dictionary-based [8], machine learning-based, and neural network-based methods. Sentiment dictionary-based methods construct a sentiment dictionary to calculate the sentiment score of the comment sentence, so it heavily relies on the predefined sentiment dictionary and external resources and does not have a stable performance in different domains. Machine learning-based methods typically train a classifier from the dataset by traditional shallow models, and they mostly need carefully designed features so that the data fed into the models can effectively represent the problem.
Meanwhile, due to their simplicity, it is difficult to mine high-
order characteristics in the language. With the rise of deep
learning, scholars have built various deep neural network-
based models to learn the dependencies within the text contents
to obtain better results.

In existing neural network-based models, most researchers
concentrate on how to construct complex network structures
to obtain better performance, while ignoring a more funda-
mental and significant challenge in ABSA task—the lack of
labelling training data [9]. Therefore, the performance
improvements are not quite obvious, and meanwhile, the
complex network structure will increase the training cost.

Regarding the issues above, Google proposes a pretrained
language model: BERT (bidirectional encoder representation
from transformers) [10]. BERT uses unlabelled data to train
the model to learn common language representations, which
provides an effective solution to above problems. BERT has a
multilayered network, and some works have discussed the
shifting roles of different layers in BERT. They show that the
middle layers of BERT encode rich hierarchical linguistic
information, with superficial features at the bottom, syntactic
features in the middle and semantic features at the top [11].

Previous studies focused on the impact of lexical and semantic
features in the output layer of BERT, while it ignored the
syntactic features in the middle layers.

In this paper, we study the impact of syntactic information
in the middle layers of BERT on the aspect-based sentiment
classification task. The syntactic features are mined by a syntac-
tic feature extraction unit (SFE) from the BERT middle layers
and are then fused with the semantic features mined from the
BERT output layer, which effectively improves the classifica-
tion accuracy.

The main contributions of this work are presented as
follows:

(1) For aspect-based sentiment classification task, we pro-
pose a new model called BERT-SFE, which is built
based on a pretraining model modified on BERT

(2) In BERT-SFE, we use the syntactic information
extracted from the BERT middle layers by a syntactic
feature extraction unit, which has contributed a lot
to mine the syntactic dependencies with in the com-
ment sentence

(3) The syntactic feature extraction unit is built based on
Star-Transformer, which can effectively mine local
and global contexts for the aspect

(4) The experimental results on three ABSA public data-
sets show that BERT-SFE performs better than exist-
ing models

2. Related Works

Aspect-based sentiment classification task is also called target-
oriented sentiment classification task. The goal is to identify
the sentiment polarity of a given aspect in a sentence, which
can be seen as a text classification problem. Traditional
methods mainly use machine learning algorithms such as
decision trees and support vector machines [12] to do the clas-
sification. These methods usually spend a lot of time to con-
struct auxiliary features or a sentiment dictionary, and their
performances are highly dependent on the quality of feature
engineering.

In recent years, with the rise of neural network, scholars
have built neural network models to learn the dependencies
between the text content and the sentiment polarity. For
example, Song et al. [13] proposed a neural network model,
using an attention-based encoder to model the context and
opinion targets, respectively. Zhang et al. [14] proposed to
add grammatical knowledge to solve the problem of long-
distance dependencies, by using grammatical dependency tree
and GAN to mine the hidden relationships between the context
and the aspect. The above neural network-based methods are
mainly to learn the correlations between text content and senti-
ment polarity by constructing a complex network structure.

However, the biggest challenge for aspect-based sentiment
classification tasks is the lack of labelled training data [9].
Improvement brought by complex network models is not obvi-
ous, and the complex networks will increase the training cost.

For above problems, researchers applied the pretrained
BERT model to aspect-based sentiment classification task. Sun et al. [16] constructed auxiliary sentences for the aspect,
which are used to fine-tune BERT to improve the performance.

KARIMI [17] uses the adversarial network to learn the general
BERT model and in-domain posttrained BERT model. Huang
et al. [18] perform external training on the pretrained BERT
model, which is combined with the graph attention network
based on the grammatical dependency tree to syntactically
perceive the model. ELECTRA [19] is a new pretrained model
that improves on the BERT model, and it can better learn the
features at the embedding layer and has shown a better capability
to capture contextual word representations [20].

The above BERT-based models focus on the impact of the output layer. However, BERT has a multilayered network. According
to the feature pyramid [21], the BERT middle layers contain a
lot of useful intermediate features. Song et al. [13] further stud-
ied the role played by different middle layers and found that
superficial features mainly reside in the bottom layers, syntactic
features are mostly in the middle layers, and semantic features
can be found at the output layers. Previous BERT-based models
ignored the syntactic features in the middle layers. In order to
better utilize this kind of information, we propose a syntactic
feature extraction unit based on Star-Transformer [15] to mine
the syntactic features in the middle layers, which are further
fused with the semantic features in the output layer to obtain
a more accurate sentiment representation, which can effectively
improve the accuracy of sentiment classification task.

3. Proposed Model

In this work, we propose a new neural network-based model
called BERT-SFE. BERT-SFE constructs a syntactic feature
extraction unit to extract syntactic features in the BERT
middle layers, which is further fused with the semantic fea-
tures extracted from the output layer. In this part, we will
make an overall explanation of the model and introduce its
component modules.
3.1. Task Formulation. Aspect-based sentiment classification task is aimed at identifying the sentiment polarity of a given aspect in sentence, which is a text classification task. Specifically, for the given comment sentence: \( S = \{ w_1, w_2, \ldots, w_n \} \), where \( n \) is the length of the sentence, \( w_i \) represents a word in comment sentence, \( t \) represents the location of the aspect in the sentence, \( w_t = \{ w_{t1}, \ldots, w_{tn} \} \), \( w_{t1} \) represents the start position of the aspect, and \( w_{tn} \) is the end. Our goal is to use tags \( y_i \in \{ \text{POS, NEG, NEU} \} \) (POS: positive; NEG: negative; NEU: neutral) to classify a given aspect.

3.2. Framework. Figure 1 is the network architecture of BERT-SFE, including a context encoding layer, a syntactic feature extraction layer, and a feature fusion layer. First, the context encoding layer uses the pretrained BERT to map words to a sequence of word embedding vectors. Secondly, in the feature extraction layer, the syntactic feature extraction unit is used to extract the syntactic features in the BERT middle layers. Finally, in the feature fusion layer, the obtained syntactic features are fused with the semantic features in the BERT output layer, and the final sentiment classification is generated.

3.3. BERT-Based Context Encoding Layer. In the context encoding layer, we use pretrained BERT as a feature extractor to map words to word embeddings. BERT uses unlabelled data to train the model and learn common language representations. Therefore, it is more expressive than RNN and LSTM.

The context encoding layer takes the comment sentence \( S = \{ w_1, w_2, \ldots, w_n \} \) containing \( n \) words as the input. First, we encode the comment sentence by concatenating the comment sentence representation (sentence_token) and the aspect representation (aspect_token) it contains. The sequence is labelled according to “CLS + sentence_token + [SEP] + aspect_token + [SEP]”. Then, the labelled data is fed into the BERT for encoding. BERT has a multilayered network, and each layer generates varying semantics. Let \( H_{\text{BERT}} \) be the complete set of features captured at different layers in BERT:

\[
H_{\text{BERT}} = \{ H_{\text{BERT}}^{1}, \ldots, H_{\text{BERT}}^{l}, \ldots, H_{\text{BERT}}^{\ell} \}.
\]  

(1)

BERT-SFE uses \( \text{BERT}_{\text{base}} \), where \( l \) represents the number of BERT layers, with value of 12; \( H_{\text{BERT}}^{l} \) is the BERT output layer. Experiments found that the last 4 BERT middle layers had a great influence on the classification results. Therefore, BERT-SFE selects the last 4 middle layers. \( H_{\text{BERT}}^{l} \) represents the features obtained in the \( l \)th layer, which contains the word embeddings of the comment sentence with contextual information encoded inside.

\[
H_{\text{BERT}}^{l} = \{ h_{l1}^{\text{CLS}}, h_{l1}^{1}, h_{l1}^{2}, \ldots, h_{l1}^{n}, h_{l1}^{\text{SEP}}, h_{l1}^{2}, h_{l1}^{\text{SEP}} \}.
\]  

(2)

3.4. Syntactic Feature Extraction Layer. In the syntactic feature extraction layer, in order to capture the syntactic features in the last 4 BERT middle layers, we construct a syntactic feature extraction unit called SFE for each layer in BERT, as shown in Figure 2, whose underlying structure is the Star-Transformer. Star-Transformer simplifies the traditional transformer by using a star network containing a central auxiliary node instead of a fully connected network or a sequence model like RNN. For a given node, it is affected by the auxiliary node and its neighboring nodes, where the auxiliary node represents global context while the neighboring nodes represent local context.

For the middle layers in BERT, the output of the Star-Transformer in the current layer and the previous layer are merged together to form the syntactic feature extraction unit SFE in the current middle layer, shown as follows:

\[
H_{\text{SFE}}^{i} = H_{\text{SFE}}^{i-1} + H_{\text{BERT}}^{i},
\]  

(3)

where \( H_{\text{SFE}}^{i} \) represents the output of the feature extraction unit SFE in the \( i \)th middle layer, \( H_{\text{BERT}}^{i} \) represents the output of the \( i \)th BERT middle layer, and \( H_{\text{SFE}}^{i-1} \) is the output of SFE from the previous middle layer.

The execution process of SFE can be described as the pseudocode below:

The auxiliary vector \( Z \) is initialized as the average of all vectors in \( H_{\text{SFE}}^{i-1} \); \( T \) represents the number of iterative trainings, which is set to 2; \( C^i \) denotes the context information for the current vector; MultiAtt(·) represents the multihread attention algorithm; ReLU(·) is the activation function; and LayerNorm is normalized function.

In previous researches on sentiment analysis, it is found that the neighboring nodes of the aspect contain more locational information [18], and the global sentence representation contains more information on syntactic relationships. In SFE, the input is \( H^{i} \) and the auxiliary vector \( Z \) is initialized as the average of all vectors in \( H^{i} \). The current vector \( h^{i} \) is updated by \( C_{li} \) and \( C_{hi} \), representing the local context, while \( Z \) is updated by all the word embeddings in the whole sentence after each training iteration, representing the global context. The local features are highlighted through the neighbouring nodes, which helps to mine the positional relationships in the middle layers; the global features are captured by the auxiliary vector \( Z \), which helps to mine the syntactic features in the BERT middle layers.

The final output generated from the syntactic feature extraction module is

\[
H_{\text{SFE}}^{\ell} = \{ H_{\text{SFE}}^{\ell-4}, H_{\text{SFE}}^{\ell-3}, H_{\text{SFE}}^{\ell-2}, H_{\text{SFE}}^{\ell-1} \},
\]  

(4)

where \( l \) represents the BERT output layer and \( H_{\text{SFE}}^{\ell} \) is the output of SFE from the last 4 middle layers.

3.5. Feature Fusion Layer. After the syntactic feature extraction layer, we get a vector sequence \( H_{\text{SFE}}^{\ell} \) containing syntactic features. At the same time, the output layer of BERT contains a large number of semantic features. The goal of the feature fusion layer is to merge syntactic and semantic features. First, \( H_{\text{SFE}}^{\ell} \) and \( H_{\text{BERT}}^{\ell} \) are linearly transformed, respectively:
\[ O_{\text{SFE}} = H_{\text{SFE}} \cdot W_{\text{SFE}} + B_{\text{SFE}}, \]
\[ O_{\text{BERT}} = H_{\text{BERT}}^{i} \cdot W_{\text{BERT}} + B_{\text{BERT}}, \]  

where \( W_{\text{SFE}} \) and \( W_{\text{BERT}} \) are weight matrices and \( B_{\text{SFE}} \) and \( B_{\text{BERT}} \) are bias vectors. Linear transformation can compress vectors and transform vector sequences into the output format of sentiment classification task.

Finally, the vectors are fused, and the fusion formula is as follows:

\[ x = O_{\text{SFE}} + O_{\text{BERT}}, \]

We use softmax function to classify the results into \( k \) categories, representing the prediction of the sentiment polarity of the aspect, shown as follows:
\[ y = \text{softmax}(x) = \frac{\exp(x)}{\sum_{k=1}^{k} \exp(x)}, \quad (7) \]

where \( k \) represents the number of sentiment categories, \( \exp() \) is exponential function, and \( y \in \mathbb{R}^k \) represents the emotional polarity distribution.

3.6. Model Training. Through formula (8), we get the probability distributions for different sentiment labels. We use the greedy decoding, which predict the sentiment polarity as the label with the highest probability. Through the cross-entropy loss function, the model is trained end-to-end to minimize the loss value.

\[ L(\theta) = -\sum_{y=1}^{k} \mathbb{1}(\bar{y} = y) \cdot \log(\hat{y} = y) + \lambda \sum \theta^2, \quad (8) \]

where \( \mathbb{1}() \) is the indicator function, \( \hat{y} \) is the predicted sentiment polarity label, \( y \) represents the real sentiment polarity label, \( \lambda \) is the adjustment parameter, and \( \theta \) is the training parameter set.

4. Experiment

4.1. Datasets and Experimental Parameter Settings. To verify the effectiveness of our proposed model, we evaluate our model on three benchmark datasets. Laptop dataset and restaurant dataset come from SemEval 2014 Task 4 [22] and Twitter comes from ACL 14 Twitter [23]. The data are labelled with three sentiment polarities: positive, negative, and neutral. Table 1 counts the number of labels in the three datasets.

We use the pretrained BERT base to encode the context. It is a 12-layered network, and the embedding dimension is 768. The coefficient \( \lambda \) is set to \( 10^{-5} \). We use Adam as the optimization function and use Dropout to solve the overfitting problem and set it to 0.1.

4.2. Evaluation Metrics. We use Accuracy and F1-scores as the evaluation metrics to compare our model with other models. The accuracy is the ratio of the predicted correct data to the entire dataset. The F1-score is a classic multilabel classification evaluation metric, which represents the overall performance of the model.

4.3. Model Comparisons. We use the following models to compare with our method:

- MemNet [24] uses multihops of attention layers on the context word embeddings for sentence representation to explicitly capture the importance of each context word.
- TD-LSTM [25] uses two LSTM networks to model the left and right contexts of the target, respectively.
- IAN [26] uses two LSTMs and an attention network to generate interactive representations of opinion targets and context.
- RAW [27] strengthens MemNet by utilizing memory for bidirectional LSTM and using a gated recurrent unit network to combine the multiple attention outputs for sentence representations.

- ASGCN [14] uses a GCN model that focuses on opinion targets and fuses syntactic knowledge to solve the problem of long-distance dependence.
- BERT base [10] is the basic BERT model, and the output layer is linearly processed for classification tasks.
- BERT-SPC [28], our benchmark model, feeds sequence “[CLS] + sentence + [SEP] + aspect + [SEP]” into the basic BERT model for aspect-based sentiment classification task.
- BERT-PT [29] uses a joint posttraining algorithm to fine-tune BERT on a mixed dataset (Amazon review and Yelp) and merges domain knowledge.
- BERT-SFE, our proposed model, introduces Star-Transformer to extract the syntactic features of the BERT middle layer and is merged with the semantic features of the BERT output layer for the aspect-based sentiment classification task.

4.4. Main Results. The main experiment results are shown in Table 2. All methods in the table use greedy decoding. It can be seen from Table 2 that, compared with the pre-trained models, the performance of the traditional neural network models is poor. TD-LSTM and MemNet are both LSTM-based models, and TD-LSTM uses two LSTM networks to model the left and right context of the target, respectively. As the syntactic structure is destroyed, the overall performance is poor and accuracy on Twitter is only 64%. MemNet has the lowest model size as it only has one shared attention layer and two linear layers, the improvement is limited. IAN and RAW are both attention-based models, and their overall performance is better than TD-LSTM and MemNet. ASGCN uses GCN to process the data on the grammatical dependency tree and mines the syntactic relationships in the sentence. It has improvements compared...
Table 3: Comparison results of BERT-Layer and BERT-SPC.

<table>
<thead>
<tr>
<th>Models</th>
<th>Laptop Accuracy</th>
<th>Restaurant Accuracy</th>
<th>Twitter Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-SPC</td>
<td>78.99</td>
<td>75.03</td>
<td>84.46</td>
</tr>
<tr>
<td>BERT-Layer</td>
<td>79.05</td>
<td>75.32</td>
<td>84.69</td>
</tr>
</tbody>
</table>

Table 4: Comparison results of BERT-Layer, BERT-LSTM, and BERT-SFE.

<table>
<thead>
<tr>
<th>Models</th>
<th>Laptop Accuracy</th>
<th>Restaurant Accuracy</th>
<th>Twitter Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Layer</td>
<td>79.17</td>
<td>75.32</td>
<td>84.69</td>
</tr>
<tr>
<td>BERT-LSTM</td>
<td>78.79</td>
<td>75.57</td>
<td>85.18</td>
</tr>
<tr>
<td>BERT-SFE</td>
<td>79.47</td>
<td>75.90</td>
<td>85.18</td>
</tr>
</tbody>
</table>

with above models, with F1 on the restaurant dataset reaching 72.19%. Although the ASGCN model constructs a complex network structure, the overall performance is still lower than the BERT-based models. The reason is that the biggest challenge of the ABSA task is the lack of labelled data; therefore, the complex network structure cannot realize their full potential. The pretrained BERT uses a large amount of unlabelled data to train the model and learn common language representations, word dependencies, together with varying granularities of syntactic and semantic information, so as to provide a better understanding on the language itself, which is particularly important for the sentiment classification task.

In the BERT-based model, BERT\textsubscript{base} only performs a linear transformation on the output layer, and the experimental results are better than non-BERT models, which prove the advantages of the pretrained BERT on small datasets. BERT-SPC is our benchmark model. It changes the way of data labelling and highlights the aspect. Therefore, the overall performance is still lower than the BERT-based models. The reason is that the biggest challenge of the ABSA task is the lack of labelled data; therefore, the complex network structure cannot realize their full potential. The pretrained BERT uses a large amount of unlabelled data to train the model and learn common language representations, word dependencies, together with varying granularities of syntactic and semantic information, so as to provide a better understanding on the language itself, which is particularly important for the sentiment classification task.

In order to verify the influence of the BERT middle layers on the sentiment classification task. We use BERT-Layer and BERT-SPC for comparative experiments, and the experimental results are shown in Table 3:

According to the controlled variable methodology, BERT-Layer only adds the BERT middle layers on the basis of BERT-SPC. It can be seen that the overall performance of BERT-Layer is better than BERT-SPC. On the Twitter dataset, accuracy and F1 are increased by 0.29% and 0.35%, respectively. The above experiments prove that the BERT middle layers can help to improve the accuracy of sentiment classification.

4.5.2. The Influence of SFE. In the above experiments, the syntactic information in the BERT middle layers is verified to be effective, but as BERT-Layer does not explicitly capture global and local context of the aspect, the performance improvement is limited. Therefore, we use BERT-LSTM and BERT-SFE to compare with BERT-Layer. The experimental results are shown in Table 4:

It can be seen from Table 4 that BERT-LSTM outperforms BERT-Layer and BERT-SFE achieves the best performance. This shows that explicitly modelling the sequential patterns in the sentences through LSTM can effectively mine the dependencies between the aspect words and their contexts. Meanwhile, the syntactic feature extraction module can effectively mine the syntactic information in the BERT middle layers, obtaining more accurate sentiment prediction for the aspect.

It can also be seen that the accuracy of BERT-LSTM is the same as BERT-SFE on the restaurant dataset. After analyzing the restaurant dataset, we found that the restaurant dataset contains a lot of colloquial sentences with poor coherence and fuzzy syntactic structures, which affects the performance of SFE. However, BERT-SFE outperforms BERT-LSTM significantly on the Twitter and laptop datasets, which proves the effectiveness of SFE.

4.5.3. The Influence of the Number of Middle Layers. In this section, we explore the influence of the number of BERT middle layers.

Clark et al. [19] found that different layers of BERT contain different gratuities of syntactic and semantic information,
and top middle layers contain more syntactic information. Therefore, we prefer to use top middle layers and constantly incorporate more bottom middle layers, to study the impact of the number of middle layers on the laptop dataset. The experimental results are shown in Figure 3.

It is found from Figure 3, when the last 4 middle layers are used, the performance is optimal. When we continue to add the middle layer, the performance decreases, as the features from shallow layers will introduce more noise. Therefore, BERT-SFE selects the last 4 middle layers for syntactic feature extraction.

5. Conclusion

Aspect-based sentiment classification task is a fine-grained sentiment analysis task, which is aimed at identifying the sentiment polarity of a given aspect in a subjective sentence. In the past, the BERT-based model only considered the impact of the output layer, while ignoring the syntactic features in the BERT middle layers. In order to better utilize syntactic information, we design an effective model called BERT-SFE. BERT-SFE constructs the Star-Transformer-based syntactic feature extraction unit SFE to extract the syntactic features in the last 4 middle layers of BERT, which is then merged with semantic features in the BERT output layer in the feature fusion layer to generate the enhanced sentiment representation of the aspect. The experimental results on three public ABSA datasets show that our model is better than existing models, which proves the effectiveness of BERT-SFE.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Humanities and Social Science Research Project of the Ministry of Education under Grant 17YJCZH187, Taishan Scholar Climbing Program of Shandong Province under Grant No. ts2090936, SDUST Research Fund under Grant No. 2015TDJH102, 2021 National Statistical Science Research Project under Grant 2021LY053, Shandong Postgraduate Education Quality Improvement Plan (No. SDYJG19075), and Shandong Education Teaching Research Key Project (No. 2021JXZ010).

References


