Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree

Kai Sun
BDBC and SKLSDE
Beihang University,China
sunkai@buaa.edu.cn

Richong Zhang*
BDBC and SKLSDE
Beihang University, China
zhangrc@act.buaa.edu.cn

Samuel Mensah
BDBC and SKLSDE
Beihang University, China
samensah@buaa.edu.cn

Yongyi Mao
School of EECS
University of Ottawa,Canada
ymao@uottawa.ca

Xudong Liu
BDBC and SKLSDE
Beihang University, Beijing, China
liuxd@act.buaa.edu.cn

Abstract

We propose a method based on neural networks to identify the sentiment polarity of opinion words expressed on a specific aspect of a sentence. Although a large majority of works typically focus on leveraging the expressive power of neural networks in handling this task, we explore the possibility of integrating dependency trees with neural networks for representation learning. To this end, we present a convolution over a dependency tree (CDT) model which exploits a Bi-directional Long Short Term Memory (Bi-LSTM) to learn representations for features of a sentence, and further enhance the embeddings with a graph convolutional network (GCN) which operates directly on the dependency tree of the sentence. Our approach propagates both contextual and dependency information from opinion words to aspect words, offering discriminative properties for supervision. Experimental results ranks our approach as the new state-of-the-art in aspect-based sentiment classification.

1 Introduction

The current explosion in digital technology in recent years has led to a vast amount of opinionated materials on the internet. In particular, individuals have expressed opinions on several aspects of products, services, blogs, and comments which are deemed to be influential, especially when making purchase decisions based on product reviews (Schouten and Frasincar, 2015). However, due to the voluminous amount of content online, sifting through reviews to learn knowledge of opinions expressed on specific aspects of a review is cumbersome. This fact has led to an increase in research in aspect-based sentiment analysis (ABSA), which aims to find scalable solutions to address the problem automatically. More specifically, ABSA involves two tasks: (1) to identify aspects of a sentence, and (2) to determine the sentiment polarity (e.g. positive, negative, neutral) expressed on a specific aspect. In this paper, we focus on the second task: aspect-based sentiment classification.

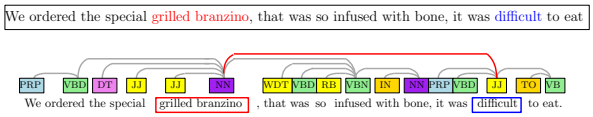


Figure 1: An example of a dependency tree where an opinion word (blue) and the specific aspect expression (red) are connected with other word tokens based on their syntactic dependencies.

With the aim to address the classification task, several methods have been developed. Majority of recent works such as (Dong et al., 2014; Tang et al., 2015; Wang et al., 2016; Chen et al., 2017; Cheng et al., 2017) have exploited neural networks due to its ability to model representations for sentences automatically. Even so, some recent methods have integrated both lexical resources with neural networks to achieve state-of-the-art performance in ABSA (Wang et al., 2018; Ouyang and Su, 2018).

Generally, we find that a dependency tree shortens the distance between the aspects and opinion words of a sentence, captures the syntactic relations between words, and offers discriminative syntactic paths on arbitrary sentences for information propagation across the tree. For instance, consider the dependency tree as depicted in Figure 1, the distance between the aspect expression *'grilled branzino'* and the opinion word *'difficult'* is shortened by a single path based on their syntactic dependencies. These properties allow neural network models to capture long-term syntactic dependencies effortlessly. Besides, dependency

*Corresponding author

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5679–5688, Hong Kong, China, November 3–7, 2019.

trees have graph-like structures bringing to play the recent class of neural networks, namely, graph convolutional networks (GCN) (Kipf and Welling, 2016). The GCN has been successful in learning representations for nodes, capturing the local position of nodes in the graph. Several AI applications such as link prediction (Schlichtkrull et al., 2018; Zitnik et al., 2018; Kong et al., 2019), semantic role labeling (Marcheggiani and Titov, 2017), and relation extraction (Zhang et al., 2018) have successfully exploited GCNs to improve representation learning.

These observations motivate us to develop a neural model which can operate on the dependency tree of a sentence, with the aim to make accurate sentiment predictions with respect to specific aspects. Specifically, we propose a convolution over a dependency tree (CDT) model which exploits a GCN to model the structure of a sentence through its dependency tree, where node (word) embeddings of the tree are initialized by means of a Bi-directional Long Short Term Memory (Bi-LSTM) network. Motivated by the recent work of (Zhang et al., 2018) in a relation extraction task, we find that the architecture of CDT allows the Bi-LSTM account for contextual information between successive words, while the GCN enhances the embeddings by modeling the dependencies along the syntactic paths of the dependency tree. Such operations allow information to be transferred from opinion words to aspect words, implying that the encoding for aspect words is sufficient for supervision in the classification task. Experimental results, including visualizations show the effectiveness of our proposed model.

2 Related Work

The performance bottleneck in the classification task of ABSA comes from modeling representations which efficiently encode the relationship between a specific aspect and the opinion words of a sentence. Most recent methods have focused on leveraging neural networks (Chen et al., 2017; Gu et al., 2018; Majumder et al., 2018; Fan et al., 2018a; Xue and Li, 2018; Huang and Carley, 2018; Zheng and Xia, 2018) which model representations automatically. Besides, in contrast to rule-based methods (Hu and Liu, 2004; Popescu and Etzioni, 2005; Ding et al., 2008; Popescu and Etzioni, 2005), neural networks are more capable of dealing with situations where opinion words are found in more complicated contexts.

Among neural network methods, some model the sentence representation using RNN variants such as LSTM and gated recurrent units (GRU). (Chen et al., 2017) handles the encoding of reviews using BiLSTM and attention networks. (Gu et al., 2018) improves the performance by considering the position of the aspect words. Similarly, (Zheng and Xia, 2018) use LSTMs to learn embeddings for the left context, right context and target phrase of sentences while considering the interactions between targets and contexts. (Majumder et al., 2018) on the other hand models the sentence representations using GRU and attention mechanisms. However convenient, these neural network based methods neglect informative resources such as dependency trees which is capable of shortening the distance between aspect and opinion words, enabling dependency information to be preserved effectively in lengthy sentences.

The state-of-the-art methods for representation learning have integrated dependency trees with neural networks. (Tai et al., 2015) proposed a tree-structured LSTM: a generalized class of LSTMs which enables the learning of dependency information between words and phrases. (Mou et al., 2015) exploit the short paths of dependency trees to learn representations of sentences using convolutional neural networks, while preserving dependency information. Motivated by such works, (Gu et al., 2018) proposed a position encoding convolutional neural network which takes into account the relative position of words and entities of a dependency tree for relation classification. Given that dependency trees can be considered as a graph, (Marcheggiani and Titov, 2017) introduced a variant of a GCN to model representations for dependency graphs in semantic role labeling tasks.

In a recent relation extraction task, (Zhang et al., 2018) extract entity-based representations via a GCN which operates on a dependecy tree. (Zhang et al., 2018) observed that stacking a GCN layer over an LSTM improves performance immensely. We follow a similar approach and propose CDT: a method which performs convolutions over a dependency tree to extract rich representations for aspect-based sentiment classification. CDT extracts a final representation for the ABSA classification task by aggregating only the aspect vec-

tors. We believe this is sufficient because the GCN componet can be interpreted as a messaging passing network which propagates information along edges. Thus, successive GCN operations allow information to be propagated across the network, and hence aspect vectors are encoded with information from opinion words which should be sufficient for supervision.

3 Convolution over Dependency Tree Model

In this section, we describe the CDT model which takes as input a dependency tree of a sentence. Node embeddings of the dependency tree are initially modeled by means of a BiLSTM, and the embeddings are further enhanced via a GCN. Finally, an aggregator is applied over the enhanced aspect embeddings to distill a dense vector embedding for the classification task. In particular, we aim to extract embeddings which encode both contextual and dependency information between a specific aspect expression and opinion words, providing supervisory signals for the aspect-based classification task.

We briefly describe the BiLSTM model, which takes as input the sentence s with n ordered word embeddings. The BiLSTM integrates context information in the word embeddings by keeping track of dependencies along the chain of words. Given an aspect-sentence pair (a, s), where $a = \{a_1, a_2, \ldots, a_l\}$ is a sub-sequence of the sentence $s = \{w_1, w_2, \ldots, w_n\}$. The sentence s has corresponding word embeddings $x = \{x_1, x_2, \ldots x_n\}$. The LSTM learns hidden state representations $\{\overrightarrow{h_1^0}, \overrightarrow{h_2^0}, \ldots \overrightarrow{h_n^0}\}$ in the forward direction on the word embeddings in x. This allows contextual information to be captured in a forward direction. In a similar fashion, a backward LSTM will learn representations $\{\overleftarrow{h_1^0}, \overleftarrow{h_2^0}, \ldots \overleftarrow{h_n^0}\}$ on x. Finally, we can concatenate the corresponding parallel representations modeled by both forward and backward LSTMs into higher dimensional representations $\{h_1^0, h_2^0, \ldots, h_n^0\}$, which contains the subsequence $\{h_{a_1}^0, h_{a_2}^0, \ldots, h_{a_l}^0\}$ corresponding to the aspect expression a. In doing so, we capture contextual information between opinion words and aspects. Besides, we integrate dependency information in the contextualized embeddings using a GCN which operates directly on the dependency tree of the sentence.

3.1 Graph Convolutional Network

The dependency tree can be interpreted as a graph G with n nodes, where nodes represent words in the sentence and edges represent syntactic dependency paths between words in the graph. The nodes of the dependency tree are given by real-valued vectors modeled by BiLSTM as described above. This structure allows a GCN to operate directly on the graph to model dependencies that exist between words. To allow the GCN to model node embeddings efficiently, we allow G to have self-loops. The GCN approach ensures that the sentence structure represented by the dependency tree is encoded efficiently, whereby the representations for nodes encode the local position of opinion words and the target words in the dependency tree.

The dependency tree G for any arbitrary sentence can be represented as an $n \times n$ adjacency matrix A, with entries A_{ij} signaling if node i is connected to node j by a single dependency path in G. Specifically, $A_{ij} = 1$ if node i is connected to node j, and $A_{ij} = 0$ otherwise. Together with node embeddings modeled by BiLSTMs, we can exploit a GCN capable of operating directly on graphs. The GCN makes efficient use of dependency paths to transform and propagate information across the paths, and update node embeddings by aggregating the propagated information. In such an operation, the GCN only considers the first-order neighborhood of a node when modeling its embeddings. However, k successive GCN operations result in the propagation of information across the k-th order neighborhood. A single node embedding update takes the form

$$h_i^{(k+1)} = \phi\left(\sum_{j=1}^{n} c^i A_{ij}\left(W^{(k)} h_j^{(k)} + b^{(k)}\right)\right), \tag{1}$$

where $h_j^{(k)}$ is the hidden state representation for node j at the k^{th} layer of the GCN, $b^{(k)}$ is a bias term, $W^{(k)}$ is a parameter matrix, c^i is a normalization constant, which we choose as $c^i = 1/d_i$. d_i denotes the degree of node i in the graph calculated as $d_i = \sum_{j=1}^{n} A_{ij}$. $\phi(\cdot)$ is a relu elementwise non-linear activation function. Note that h_i^0 represent the initial embeddings modeled by a BiLSTM, and $h_i^{(k+1)}$ is the final output for node i at layer k.

In extracting a final embedding for the classifi-

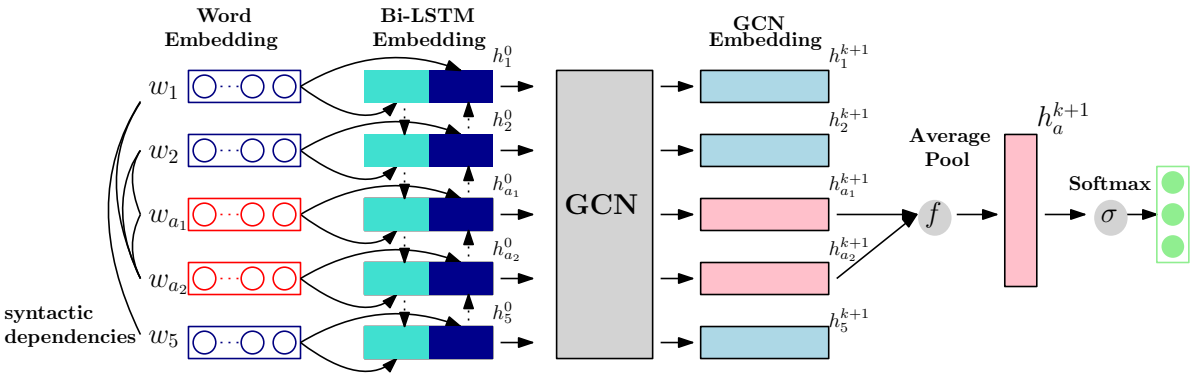


Figure 2: Overview of the CDT model based on the sentence $s = [w_1, w_2, w_{a_1}, w_{a_2}, w_5]$, where $[w_{a_1}, w_{a_2}]$ is the specific aspect expression in s, and k is the number of GCN layers.

cation task, we exploit a simple aggregator. For our framework, we choose an average pool which aggregates information over the aspect vectors. We choose to aggregate only the aspect vectors because we believe that these vectors encode contextual and dependency information owing to the BiLSTM and the GCN respectively. The BiLSTM and the GCN can be interpreted as message passing networks. Specifically, the BiLSTM allow aspect words of an arbitrary sentence to be contextualize, while the GCN finds the local position of aspect words in the syntactic dependency tree. The local position within the dependency tree encodes dependency information of a word with respect to its neighbors. As a result, the BiLSTM and the GCN allow embeddings for aspect words to have discriminative features, providing supervisory signals for the classification task. Moreover, we perform an average pool to retain most of the information in the aspect vectors. The pool operation over the aspect vectors takes the form of

$$h_a^{(k+1)} = f\left(\{h_{a_1}^{k+1}, h_{a_2}^{k+1}, \ldots, h_{a_l}^{k+1}\}\right), \quad (2)$$

where $f(\cdot)$ is an average pool function applied over the enhanced aspect vectors. We present an overview of the model architecture in Figure 2 based on an example sentence input.

4 Model Training

The aspect-based representation $h_a^{(k+1)}$ is passed to a fully connected softmax layer σ whose output is a probability distribution over the different sentiment polarities. The model is trained end-to-end through a backpropagation, where the objective function to be minimized is the cross entropy error defined as

$$J(\theta_1, \theta_2) = -\sum_{(a,s)\in D}\sum_{c\in C} y_c((a,s)) \log \hat{y}_c((a,s)), \quad (3)$$

where D is a collection of aspect-sentence pairs, C is the collection of distinct sentiment classes, $y_c((a,s))$ is the ground truth for (a,s) which takes the value of either 1 or 0. Besides, (a,s) can belong to only one sentiment class. Hence $y_c((a,s)) = 1$ indicates that the ground truth sentiment class for (a,s) is c. $\hat{y}_c((a,s))$ is the model's prediction for (a,s). θ_1, θ_2 are trainable parameters for the BiLSTM and GCN respectively.

5 Experiment

In this section, we conduct experiments to validate our model which we denote as CDT on benchmark datasets. We also present restricted versions of our model denoted as ASP-BiLSTM and ASP-GCN. Unlike our main model, ASP-BiLSTM only exploits BiLSTM to model contextual information with respect to a specific aspect expression, while ASP-GCN exploits a GCN to model dependencies between words. Both models extract a final embedding on the aspect vectors. We propose these two models to observe the performance of GCN and BiLSTM, as well as the performance when we stack a GCN on a BiLSTM which forms the CDT model. To distinguish CDT as the new state-of-the-art in aspect-based sentiment classification, we compare CDT with several well established models, showing that CDT outperforms the very recent models in the classification task. In particular, we perform case studies with visualizations to verify our approach of aggregating only aspect vectors for the final embedding. We further present visualizations on case examples showing how GCN

improves on a simple BiLSTM model.

5.1 Datasets

We evaluate the performance of our model on SemEval 2014 (Pontiki et al., 2014), which consists of restaurant reviews (Rest14) and laptop reviews (Laptop14). We also evaluate our model on SemEval 2016 [1] containing restaurant reviews (Rest16). Experiments are also performed on a collection of tweets from Twitter provided in the works of (Dong et al., 2014). We summarize the statistics of the datasets in Table 1.

Dataset	Positive		Neutral		Negative	
	train	test	train	test	train	test
Rest14	2164	727	637	196	807	196
Laptop14	976	337	455	167	851	128
Rest16	1657	611	101	44	748	204
Twitter	1507	172	3016	336	1528	169

Table 1: Distribution of samples by class labels on benchmark datasets

5.2 Implementation and parameter settings

For fairness in model comparation, we use similar parameters in compared models. Specifically, we exploit 300-dimensional Glove vectors (Pennington et al., 2014) for the word embeddings, as well as a 30-dimensional part-of-speech (POS) embeddings, 30-dimensional position embeddings, which is used to identify the relative position of each word with respect to the aspect in the sentence. We concatenate both word, POS and position embeddings, and learn a 50-dimensional BiLSTM embeddings for each word. The GCN operates on the dependency tree of the sentence to enhance the BiLSTM embeddings. All sentences are parsed by the Stanford parser. [2] To encourage the GCN to model dependencies between words, we randomly dropout 10% of neurons per layer, and about 0.7 at the input layer. The GCN model is trained for 100 epochs with batch size 32. We use the adam optimizer with learning rate 0.01 for all datasets. The code for our model is found on the Github page [3].

5.3 Compared Prior Art

As a baseline, we include CNN and LSTM models, which learn representations from both word embeddings and position embeddings. We denote these models as CNN+Position and LSTM+Position. We also include a CNN baseline method which exploits an attention mechanism to model the relation between aspect words and context words. We denote this model as CNN+ATT. These models extract a final embedding by aggregating all learned embeddings using an average pool. In particular, we compare our proposed model with very recent models on the benchmark datasets. The models we consider include,

- TNet (Li et al., 2018a): In this work, BiLSTM embeddings are transformed into target-specific embeddings, and a CNN model is used to extract a final embedding.
- PRET+MULT (He et al., 2018b): A multi-task framework based on LSTMs is proposed to transfer knowledge from a document-level model task to an aspect-level model task.
- SA-LSTM-P (Wang and Lu, 2018): This work first learn embeddings using BiLSTM and model structural dependencies between words by means of a segmentation attention mechanism.
- LSTM+SynATT+TarRep (He et al., 2018a): This method models target representation as a weighted sum of aspect embeddings, and models the syntactic structure of the sentence using an attention mechanism.
- MGAN (Fan et al., 2018b): A BiLSTM is exploited to capture contextual information in the sentence, while a multi-grained attention mechanism is proposed to extract an embedding which effectively captures the interaction between the aspect and the context.
- MGAN (Li et al., 2018b): This work integrates an alignment mechanism in a multi-task model comprising of an aspect-term task and an aspect-category task to effectively extract aspect-specific representations.
- HSCN (Lei et al., 2019): A model is proposed to capture interactions between the context and target, select target words and extract target-specific contextual representation, while measuring the deviation between target-specific contextual representation and target representations.

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

[2] https://stanfordnlp.github.io/CoreNLP/

[3] https://github.com/sunkaikai/CDT_ABSA

Model	Rest14		Laptop		Twitter		Rest16	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
SOTA	81.60	71.91	76.54	71.75	74.97	73.6	85.58	69.76
CNN+Position	79.37	68.64	72.73	68.28	72.69	70.92	84.63	64.75
LSTM+Position	77.59	67.05	70.06	64.46	71.39	69.45	83.47	62.69
CNN+ATT	79.46	69.44	70.53	64.27	73.12	71.01	84.28	60.86
TNet (Li et al., 2018a)	80.79	70.84	76.54	71.75	74.97	73.6	-	-
PRET+MULT (He et al., 2018b)	79.11	69.73	71.15	67.46	-	-	85.58	69.76
SA-LSTM-P (Wang and Lu, 2018)	81.60	-	75.1	-	69.0	-	-	-
LSTM+SynATT+TarRep (He et al., 2018a)	80.63	71.32	71.94	69.23	-	-	84.61	67.45
MGAN (Fan et al., 2018b)	81.25	71.94	75.39	72.47	72.54	70.81	-	-
MGAN (Li et al., 2018b)	81.49	71.48	76.21	71.42	74.62	73.53	-	-
HSCN (Li et al., 2018b)	77.8	70.20	76.1	72.5	69.6	66.1	-	-
ASP-BiLSTM	80.95	72.38	74.22	69.35	73.66	72.32	85.12	66.92
ASP-GCN	81.30	73.18	74.53	69.78	70.91	69.07	81.85	61.2
CDT	**82.30**	**74.02**	**77.19**	**72.99**	74.66	**73.66**	**85.58**	**69.93**

Table 2: Performance comparison on different models on the benchmark datasets. The best performance are bold-typed.

5.4 Performance Comparison

In this section, we compare model performance of recent methods with CDT, ASP-BiLSTM and ASP-GCN. We implement and report results for the baseline methods CNN+Position, LSTM+Position and CNN+ATT, and report the results in the original paper for the recent models under comparison. The classification results are shown in Table 2.

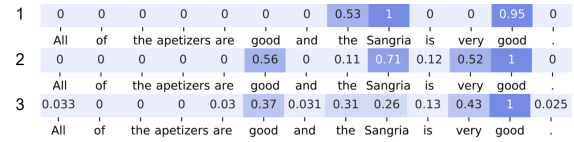
From the table, we find that CDT generally ouperforms all models for the different datasets, while having a slight accuracy performance degradation of 0.31 on the twitter dataset for the TNet model. The difference between TNet and CDT is not really significant. Hence it is fair to conclude that both models are competitive on the Twitter dataset. Even with simple architectures, we find that ASP-BiLSTM and ASP-GCN have competitive performance with the recent models on benchmark datasets. Particularly, ASP-GCN outperforms the models on the Rest14 dataset.

ASP-BiLSTM, ASP-GCN and CDT extract final representations from only the aspect vectors. Based on the performance, it seems as a sufficient technique for the classification task. We believe that the aspect vector is encoded with context and dependency information from the context and structure of the sentence by means of the BiLSTM and the GCN. The BiLSTM and GCN can be regarded as message passing networks, propagating information along a chain of sequence of words(BiLSTM) or along syntactic dependency path(GCN). Due to the fact that relevant information is passed to the aspect words, a simple average pool is all we need to retain information relevant to the classification task. Note that the information propagated in the network is learned therefore only weighed information is encoded within the aspect words.

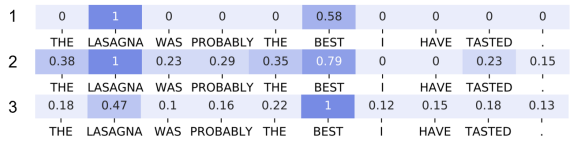
5.5 GCN Performance

We conduct an experiment to demonstrate that the performance of our proposed models, namely CDT and ASP-GCN, depend on the number of layers of the GCN. We perform this experiment on the Rest14 dataset and present the result in Figure 4.

In our experimentation, we find that as we increase the number of layers the accuracy performance increase to an extent. In particular, ASP-GCN increase in model performance over 6 layers of the GCN. The performance becomes unstable after the 6-th layer. Since GCN passes information in the local neighborhood of any node, successive operations on the dependency tree allows ASP-GCN to pass information to the furthest node. The problem of overfitting takes effect when the layers rises beyond a threshold, explaining the accuracy curve after the 6-th layer in the figure. Another important observation is the convergence of accuracy performance of the ASP-BiLSTM and ASP-GCN at the 6-th layer. Note that ASP-BiLSTM only captures contextual information while ASP-GCN captures dependency information. However, both models converge in performance at the 6-th layer. Taking advantage of the GCN and the BiLSTM we expect to improve performance, capturing both context and dependencies with respect to the aspect expression. As seen in the accuracy curve of CDT, the GCN integrates dependency informa-



(a) Case Example 1: aspect word is 'Sangria'



(b) Case Example 2: aspect word is 'LASAGNA'

Figure 3: Word relevance scores with respect to the final embedding of ASP-GCN. Number of layers for ASP-GCN is 1, 2, 3 for row 1, row 2, and row 3 respectively in both examples.

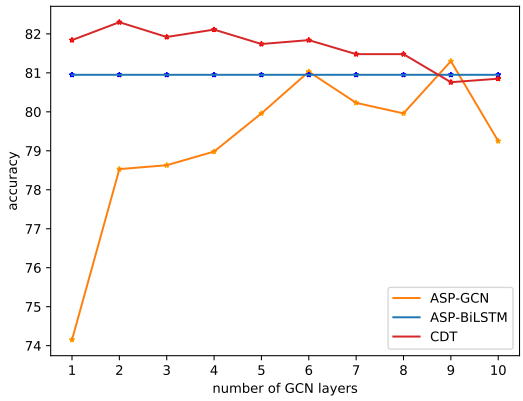


Figure 4: Accuracy curves for ASP-GCN, ASP-BiLSTM and CDT on the Rest14 dataset

tion in the contextualized embeddings to improve accuracy performance over just 2 layers, reducing the number of GCN layers needed.

5.6 Mask Experiment

Our primary assumption was that the aspect embeddings learned by our GCN model contains sufficient information necessary for the classification task. Based on this assumption we aggregate only the aspect embeddings using a max pool with the aim to retain most of the information. To verify this assumption, we trace from the input embeddings to the final embedding. We propose a mask method designed to estimate the relevance of a word with respect to the final embedding, and perform mask experiments using ASP-GCN.

The mask method works as follows. First, we follow through the conventional procedure to extract a final embedding h_s for a given sentence s using ASP-GCN. We perform a subsequent run of ASP-GCN on the same sentence s to extract a final embedding, but in this instance we conceal a specific input word w. We conceal w by mapping it to the zero vector before applying ASP-GCN on s. As a result a final embedding $h_{(s\backslash w)}$ is generated for s. If $h_s = h_{(s\backslash w)}$, the word w has no impact on the representation h_s. In other words, h_s does not capture w or no information flows from w to h_s. To this end, we can define a score function to estimate the relevance of w on h_s. We define the score function for w as

$$\gamma(w, s) = m \sum_{i=1}^{d} |h_s^i - h_{(s\backslash w)}^i| \quad (4)$$

where d is the dimension of the final embedding distilled by ASP-GCN, m is a normalization constant which we choose as $m = \max_{w \in s} \gamma(w, s)$. Generally, the final embedding should capture information on opinion words with respect to the target aspect. Hence, we expect to score high values for opinion words. Consider the scores for words shown in Figure 3, we find that γ scores high values for opinion words as we increase the number of layers of ASP-GCN, while reducing scores on irrelevant words. Implying that the final embedding captures information from opinions. The results as seen in these case examples convinces us that the final embedding distilled by our model captures relevant information necessary for the classification task.

5.7 Case Study

In this section we study the behaviour of ASP-BiLSTM, TNet and CDT on case examples. To this end we present visualizations showing the attention these models place on words. For a good model, we expect the model to attend to words which influence the sentiment inferred on a specific aspect.

From Table 2 and Figure 4, it is clear that GCN complements the BiLSTM to improve model performance. This means that the BiLSTM can identify opinion words within the context with respect to a specific aspect. However, in some complicated contexts, it might perform poorly. But the GCN can build upon BiLSTM to attend to the correct opinion words by leveraging the dependencies among words. Consider the case example shown

Figure 5: Attention visualization for ASP-BiLSTM (1st row), TNet(2nd row) and CDT (3rd row) for the aspect word 'Sangria'

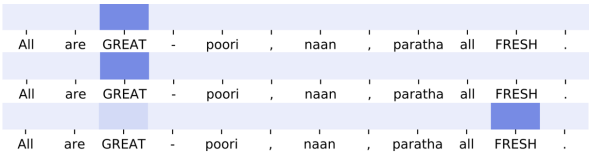


Figure 6: Attention visualization for ASP-BiLSTM (1st row), TNet (2nd row) and CDT (3rd row) for the aspect word 'paratha'

in Figure 5, ASP-BiLSTM was clever to know that the word 'good' is an opinion word with respect to the aspect 'Sangria'. But ASP-BiLSTM failed to identify whether 'good' on the far left is associated to the 'Sangria' or 'good' on the far right is associated to 'Sangria'. Interestingly, we find that the GCN could analyze this further through the dependencies between words to identify that it is the 'good' on the far right. TNet on the other hand measures the association between 'Sangria' and 'good' in both directions to identify the correct 'good'.

In Figure 6, even though the BiLSTM is able to identify the opinion word 'GREAT' which expresses an opinion on the aspect 'parathra', CDT is able to capture the opinion word 'FRESH' which directly expresses the sentiment towards the aspect. However, from the visualization is easily observed that CDT still attends to 'GREAT'. This suggest that the GCN is able to model the importance of the words with respect to the aspect, placing larger weights to words directly expressing the opinion on the aspect. At the same time, TNet misses the opinion word 'FRESH' and places attention to the word 'GREAT' just like ASP-BiLSTM.

In the case example shown in Figure 7, we find that ASP-BiLSTM places small attention on the opinion word 'BEST' which expresses the sentiment on the aspect word 'LASAGNA', while focusing its attention on 'WAS PROBABLY' which is not meaningful alone. Interestingly, CDT builds upon this little information and rely on the dependencies between the words through the dependency tree to learn that 'BEST' is the correct word to attend to. Similar to ASP-BiLSTM, TNet misses the important word 'BEST' and places attention to 'WAS PROBABLY'. This result suggest that TNet heavily depends on the representations modeled by its BiLSTM layer, while CDT considers other information such as the dependencies among words to accurately identify words which expresses opinions on specific aspects.

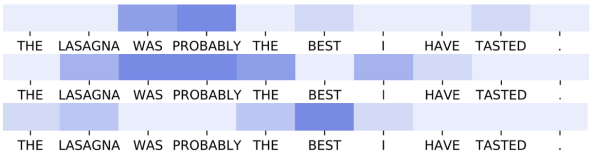


Figure 7: Attention visualization for ASP-BiLSTM (1st row), TNet (2nd row) and CDT (3rd row) for the aspect word 'LASAGNA'

6 Conclusion

Modeling representations for aspect-based sentiment classification generally require capturing informative words which express the sentiment inferred on the target aspect. Leveraging neural networks are highly desirable for representation learning. BiLSTM-based models have been successful to capture contextual information in prior works.

In this paper, we integrate a GCN with a simple BiLSTM model, with the aim to capture structural and contextual information of sentences. We have shown that the GCN successfully performs convolutions on the dependency tree to refine BiLSTM embeddings. Experimental results with visualizations support our argument on the extraction of a final embedding based on only the aspect vectors. In fact, the model we propose is simple and outperforms more complex and recent models tackling the same problem.

Acknowledgment

This work is supported partly by China 973 program (No. 2015CB358700), by the National Natural Science Foundation of China (No. 61772059, 61421003), by the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), by State Key Laboratory of Software Development Environment (No. SKLSDE-2018ZX-17) and by the Fundamental Research Funds for the Central Universities.

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 97–106.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240.

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 *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 49–54.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018a. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3433–3442.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018b. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3433–3442.

Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song. 2018. A position-aware bidirectional attention network for aspect-level sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 774–784.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018a. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1121–1131.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018b. Exploiting document knowledge for aspect-level sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 579–585.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Binxuan Huang and Kathleen Carley. 2018. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Fanshuang Kong, Richong Zhang, Hongyu Guo, Samuel Mensah, Zhiyuan Hu, and Yongyi Mao. 2019. A neural bag-of-words modelling framework for link prediction in knowledge bases with sparse connectivity. In *WWW 2019 : The Web Conference*.

Zeyang Lei, Yujiu Yang, Min Yang, Wei Zhao, Jun Guo, and Yi Liu. 2019. A human-like semantic cognition network for aspect-level sentiment classification. *AAAI*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 946–956.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, Xin Li, and Qiang Yang. 2018b. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. *CoRR*, abs/1811.10999.

Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. Iarm: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3402–3411.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. *arXiv preprint arXiv:1504.01106*.

Zhifan Ouyang and Jindian Su. 2018. Dependency parsing and attention network for aspect-level sentiment classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 391–403.

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, 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.

Ana Maria Popescu and Orena Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Bailin Wang and Wei Lu. 2018. Learning latent opinions for aspect-level sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5537–5544.

Yanyan Wang, Qun Chen, Xin Liu, Murtadha H. M. Ahmed, Zhanhuai Li, Wei Pan, and Hailong Liu. 2018. Senhint: A joint framework for aspect-level sentiment analysis by deep neural networks and linguistic hints. In *WWW '18 Companion Proceedings of the The Web Conference 2018*, pages 207–210.

Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. *arXiv preprint arXiv:1805.07043*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. *empirical methods in natural language processing*, pages 2205–2215.

Shiliang Zheng and Rui Xia. 2018. Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention. *arXiv preprint arXiv:1802.00892*.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.