Replicate, Walk, and Stop on Syntax: an Effective Neural Network Model for Aspect-Level Sentiment Classification

Yaowei Zheng,[1,2] **Richong Zhang,**[1,2*] **Samuel Mensah,**[1,2] **Yongyi Mao**[3]
[1]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Beijing Advanced Institution on Big Data and Brain Computing, Beihang University, Beijing, China
[3]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
hiyouga@buaa.edu.cn, zhangrc@act.buaa.edu.cn, samensah@buaa.edu.cn, ymao@uottawa.ca

Abstract

Aspect-level sentiment classification (ALSC) aims at predicting the sentiment polarity of a specific aspect term occurring in a sentence. This task requires learning a representation by aggregating the relevant contextual features concerning the aspect term. Existing methods cannot sufficiently leverage the syntactic structure of the sentence, and hence are difficult to distinguish different sentiments for multiple aspects in a sentence. We perceive the limitations of the previous methods and propose a hypothesis about finding crucial contextual information with the help of syntactic structure. For this purpose, we present a neural network model named RepWalk which performs a replicated random walk on a syntax graph, to effectively focus on the informative contextual words. Empirical studies show that our model outperforms recent models on most of the benchmark datasets for the ALSC task. The results suggest that our method for incorporating syntactic structure enriches the representation for the classification.

Introduction

Aspect level sentiment classification (ALSC) is a fundamental task in sentiment analysis (Pang, Lee, and others 2008; Liu 2012), which tries to infer the sentiment polarity of a sentence toward a specific aspect term. Compared to document-level or sentence-level sentiment classification, the main challenge of this task is to distinguish the different sentiments toward each aspect term when there are multiple aspects in the sentence. As a concrete example shown in Figure 1, the sentence expresses a positive sentiment on the food of a restaurant but a negative sentiment on its service.

Recurrent Neural Networks (RNNs), equipped with attention mechanism and memory module, is the most commonly used technique for ALSC (Wang et al. 2016b; Tang, Qin, and Liu 2016; Ma et al. 2017; Chen et al. 2017). These methods try to capture the semantic relationship between each contextual word and the aspect term along a chain of words. However, these models still experience the problem of long-term dependencies in the ALSC task, and cannot efficiently identify the difference between multiple aspects existing in a single sentence.

*Corresponding author: zhangrc@act.buaa.edu.cn

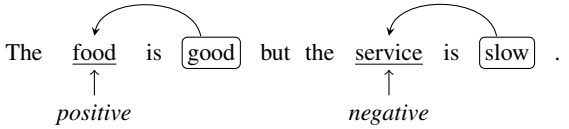


Figure 1: An example of a restaurant review with two aspect terms having different sentiments.

To cope with the aforementioned limitation, the syntactic structure has been introduced to the ALSC problem (Dong et al. 2014; He et al. 2018a; Zhang, Li, and Song 2019). The dependency tree is an important syntactic structure which establishes relationships between "head" words and words which modify those heads. Therefore, it can easily model the syntactic interrelationship between each contextual word and the aspect term.

Several recent studies explore techniques of incorporating syntactic structure, such as the dependency tree into neural network models. Empirical results suggest that leveraging syntactic structure can improve the performance of neural network models, but these models still suffer from several drawbacks. AdaRNN (Dong et al. 2014) learns to propagate sentiments of words toward the target over a converted dependency tree. However, their method destructs the original tree structure and cannot make good use of the natural structural information of the dependency tree. On the other hand, He et al. (2018a) assumes that the informative words are close to the aspect term in the dependency tree and defines an attention window to focus on closer words. This approach is not optimal and might lead to loss of information since it is based on a user-defined window size.

In summary, the important question here is how the syntactic structure can be effectively leveraged for ALSC. We investigate the structure of the dependency tree and find that only a specific subtree in the whole dependency tree is relevant to the sentiment expressed on the aspect term. For example, Figure 2 shows a dependency tree for the sentence *"The food is good but the service is slow."*. It can easily be seen that the word *"good"* in the subtree *"The food is good"* plays an important role in identifying the sentiment polarity

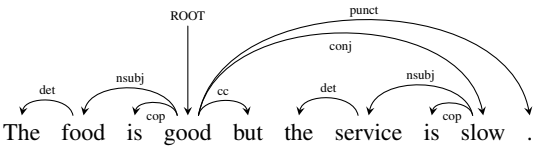


Figure 2: An example of a dependency tree.

on the aspect term *"food"* compared to the word *"slow"* in the subtree *"the service is slow"*. Based on this observation, we assume that the essential information for the ALSC task is associated with some informative words on such a subtree of the dependency tree. We formulate this assumption in Hypothesis 1.

Hypothesis 1 *The contextual features of the informative words on the subtree of the dependency tree may contribute significantly to identify the sentiment expressed on the specific aspect in the sentence.*

To validate this hypothesis, we present a novel neural network model named RepWalk, which performs a replicated random walk on a syntax graph for ALSC. This model improves the sentence representation by aggregating the crucial contextual features with the help of syntactic structures. The syntax graph induced from the dependency tree maintains the original structural information of the tree. Meanwhile, the replicated random walk process is able to focus on the informative contextual words by activating the edges in the syntax graph. We demonstrate the effectiveness of our model on incorporating the syntactic structure and validate the hypothesis through extensive experiments.

The main contribution can be summarized as follows:

- We assume that the essential information for the ALSC task is associated with the informative contextual words on the specific subtree in the whole dependency tree and propose a hypothesis on this assumption.
- Based on our hypothesis, we present RepWalk, a novel neural network for aspect-level sentiment classification. The RepWalk performs a replicated random walk on the syntax graph for learning a better sentence representation.
- We conduct extensive experiments on four widely used benchmarks to verify the effectiveness of our model. Experimental results demonstrate that our approach achieves better results compared to other strong competitors.

Related Work

Several approaches have been proposed to address the problem of ALSC in recent years. The works can be divided into three trends: the rule-based methods, the semantic-based methods, and the syntactic-based methods. In this section, we describe these works for the ALSC task.

Rule-based methods highly depend on extensive handcrafted features, which leads to error propagation in these methods, and will therefore hinder the performance of the ALSC task. Kiritchenko et al. (2014) propose a rule-based method which utilizes a Support Vector Machine (SVM) on n-gram features, parse features and lexicon features. The set of rules used to extract features in this work resulted in the best performance for the classification task in SemEval 2014. Jiang et al. (2011) generate target-dependent and target-independent features for the ALSC task using NLP tools. Similar to Kiritchenko et al. (2014), these features are also derived from handcrafted rules and may surfer from error propagation as a result of human ingenuity.

Semantic-based methods automatically learn representations for textual data with the help of the expressive power of neural network models. Recent works such as Tang et al.; Xue and Li (2016; 2018) have successfully adopted neural networks for ALSC. In particular, Wang et al.; Ma et al.; Fan, Feng, and Zhao (2016b; 2017; 2018) incorporate attention mechanism to LSTM which can help the model explicitly capture the relevance of the aspect term and the contextual words. Wang and Lu (2018) propose a segmentation attention-based LSTM model with a linear-chain conditional random field (CRF) layer, which simulates the human's process of sentiment inference, and Lei et al. (2019) further introduce human cognitive behaviors into this task. Several approaches (Li et al. 2018; Tang et al. 2019) adopt a transformation structure to improve the accuracy and efficiency of the neural network. On the other hand, it is worthy to mention that a separate class of neural architectures, known as MemNN or End-To-End Memory Networks has also been used for aspect-level sentiment analysis (Tang, Qin, and Liu 2016; Chen et al. 2017; Wang et al. 2018). Furthermore, the power of transfer learning and multi-task learning has been demonstrated in recent works (He et al. 2018b; Li et al. 2019; Chen and Qian 2019).

Syntactic-based methods integrate the dependency tree of sentences in neural network models with the help of modern high-speed dependency parsers (Chen and Manning 2014). AdaRNN (Dong et al. 2014) learns to propagate the sentiment of words to the target using recursive neural networks over the syntactic structure of sentences. He et al. (2018a) introduces a syntax-based attention mechanism based on the assumption that informative contextual words are close to the aspect term in the dependency tree. Hence, it defines an attention window to focus on such words. Zhang, Li, and Song (2019) propose a proximity-weighted convolution network for a similar purpose. In this paper, we build upon these works to integrate syntactic structure in an effective way.

Problem Statement

The ALSC task can be formulated as follows. Given a sentence $\boldsymbol{x} = \{w_1, w_2, \cdots, w_n\}$ consisting of a series of n words, and a specific aspect $\boldsymbol{a} = \{w_{a_1}, \cdots, w_{a_m}\}$ which is a span in $\boldsymbol{x}$, ALSC aims at predicting the sentiment polarity of sentence $\boldsymbol{x}$ relative to the specific aspect $\boldsymbol{a}$. This requires us to learn a sentence representation which captures the relationship between contextual words w_i and the specific aspect term $\boldsymbol{a}$ for sentence $\boldsymbol{x}$.

The quality of this sentence representation is crucial for the classification performance, which can be improved by incorporating the syntactic structure. Given a dependency tree of a sentence $\boldsymbol{x}$ which can be converted by utilizing recently advanced parsing technology (Chen and Manning

2014). Each word w_i has a unique head $w_j = h(w_i)$ in the tree except the root node. Moreover, each edge within the tree is assigned a dependency label to distinguish different dependency types of the edges.

Based on our hypothesis, we can leverage the contextual features of the informative words on the subtree of the dependency tree for a better sentence representation. To this end, we formulate our problem on effectively aggregating the important contextual features by finding such words.

Our Model

We propose a replicated random walk on the syntax graph to address the problem. In the following sections, we will present the construction of a syntax graph and illustrate the process of the replicated random walk on the syntax graph.

The construction of a Syntax Graph

To prepare for the learning process, we construct a *syntax graph* which maintains the original structural information of the dependency tree. It ensures that the paths connecting the aspect term (root node) and contextual words are unique, and all other words in the context are transitively dependent on the aspect term. We aim to find informative contextual words along these paths in the graph.

Before constructing the syntax graph, the dependency tree is converted to an aspect-rooted dependency tree. We traverse from the first word in the aspect term to each word in the dependency tree, at the same time reversing directions of some edges to allow our traversal. The conversion allows the root node of the converted tree to be the first word in the aspect term, with the directions of some edges reversed. We label reversed edges with different notations because we believe that they carry different properties. For example, *r#nsubj* is labeled on the edge that has the reversed direction to an edge labeled *nsubj*. The syntax graph is then constructed by adding a stop node to each original node of the tree by means of an extension edge labeled by *ext*. The stop node is a dummy node configured to support the learning process, which represents the termination of the paths from the aspect term to each word. A stop node is denoted by s_i if it attaches to the word w_i.

The syntax graph can be interpreted as a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which corresponds to a sentence $\boldsymbol{x}$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of directed edges in the graph. Let $\mathcal{T}$ and $\mathcal{C}$ be the sets of original nodes and stop nodes in the syntax graph. We also have a mapping function $\pi(t) : \mathcal{T} \rightarrow \mathcal{C}$ maps each original node to a corresponding stop node. Moreover, we can get the unique head node $h(t) \in \mathcal{T}$ for each original node $t \in \mathcal{T}$, except the root node in the syntax graph. Let $\mathcal{R}$ be a set of distinct edge labels (e.g. *det, nsubj, r#nsubj, ext, etc.*). The set of edges from the original node t to the stop node $\pi(t)$ is $\mathcal{E}_{\mathcal{C}} = \{t \in \mathcal{T}, r \in \mathcal{R} : (t, r, \pi(t))\}$, and the set of edges from the head node $h(t)$ to the original node t is $\mathcal{E}_{\mathcal{T}} = \{t \in \mathcal{T}, r \in \mathcal{R} : (h(t), r, t)\}$. All nodes in the syntax graph are the union of the two sets, $\mathcal{V} = \mathcal{T} \cup \mathcal{C}$. Similarly, edges in the graph are $\mathcal{E} = \mathcal{E}_{\mathcal{T}} \cup \mathcal{E}_{\mathcal{C}}$. For easy understanding, an example of the aspect-rooted dependency tree and the syntax graph can be found in Figure 3.

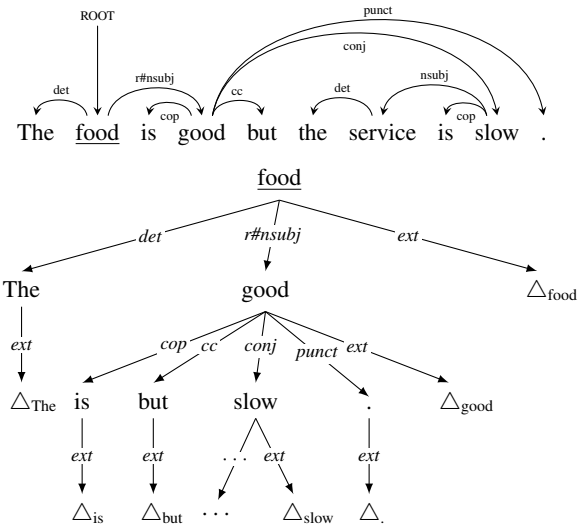


Figure 3: A concrete example of the aspect-rooted dependency tree (up) and the corresponding syntax graph (below). The original text is *"The food is good but the service is slow."*, which expresses a positive sentiment on the aspect term *"food"*. Notation "△" indicates a stop node.

Replicated Random Walk on the Syntax Graph

To effectively find the informative contextual words with the help of the syntax graph, we draw on the principle of the random walk for the ALSC task. The learning problem can be understood on the behavior of an agent traversing from the root node of the syntax graph, replicating itself on each node, with the aim to traverse on all available routes reaching a stop node with some probability. The word w_i will be highlighted if a replica manages to traverse onto the stop node s_i along the path in the syntax graph. We refer to such a traversal as a *replicated random walk*. The replicated random walk process contains three parts: Replicate, Walk and Stop, and the details are illustrated below.

Replicate Process To distinguish multiple aspects in the same sentence, the travel route starts with the current aspect word node. Because there may be multiple paths connecting the informative words on the subtree of the dependency tree, only one agent cannot attend to all paths. To address this problem, the agent replicates into a total of d copies at each node, where d is the number of the downstream edges of this node. Then the replicas turn to each downstream edge and embark on their itinerary.

Walk Process Each replica walks along its downstream edge. It arrives at the next node if the edge is activated or dies at the current node. We compute a probability for each edge activation by measuring how likely the edge and the informative contextual words belong to the same subtree of the dependency tree. More specifically, we take the edge information, including node representations and edge type to compute such a probability.

Let $\boldsymbol{q}_j \in \mathbb{R}^{d_q}$ be the representation for the node $j \in \mathcal{V}$.

For any given edge $e = (u, r, v) \in \mathcal{E}$ which connects the node $u \in \mathcal{V}$ to $v \in \mathcal{V}$ with a distinct edge label $r \in \mathcal{R}$, where the embedding for r is $\boldsymbol{\theta}_r \in \mathbb{R}^{d_r}$, we define a function $p(e)$ mapping each edge e to a probability value:

$$p(e) = \sigma\left(\begin{bmatrix}\boldsymbol{q}_u \\ \boldsymbol{q}_v\end{bmatrix}^T W_p \boldsymbol{\theta}_r + b_p\right). \quad (1)$$

where σ denotes the *sigmoid* activation function, and $W_p \in \mathbb{R}^{2d_q \times d_r}$ and $b_p \in \mathbb{R}$ are learned parameters.

Stop Process After walking through the travel routes from the root node of the syntax graph, each replica ends on a stop node in the graph with a probability or dies halfway. This probability is the likelihood of the word being highlighted on the subtree of the dependency tree, and we define it as the weight of the word w_i. The weight α_i is computed by multiplying the probabilities of the edges along the unique path from the root node to each stop node s_i, which is a child node of w_i in the syntax graph. Thus a replica reaches a stop node s_i with probability α_i or dies with probability $1 - \alpha_i$. Besides, the weight α_i is always zero for any word in the aspect term because we assume that no sentiment is expressed in the aspect term. Now we denote $\mathcal{E}_{s_i}$ as the set of edges in the unique path from the root node to the stop node s_i. Formally, the weight α_i can be calculated as:

$$\alpha_i = \begin{cases} 0, & a_1 \le i \le a_m \\ \prod_{e \in \mathcal{E}_{s_i}} p(e), & \text{otherwise} \end{cases} \quad (2)$$

Finally, using the node representation $\boldsymbol{q}_i$ of node w_i, we can compute a representation for the sentence based on the node representations and the calculated weights:

$$\boldsymbol{o} = \sum_{i=1}^{n} \alpha_i \boldsymbol{q}_i \quad (3)$$

Overall Structure

We explore an RNN-based approach to contextual representation that aims to model the semantic associations within the contextual words in sentence $\boldsymbol{x}$ and the aspect term $\boldsymbol{a}$. In dealing with representation learning for the sentence $\boldsymbol{x}$, we begin by mapping each word w_i in $\boldsymbol{x}$ into its embedding vector $\boldsymbol{g}_i \in \mathbb{R}^{d_w}$. At this point, we employ GRU networks (Cho et al. 2014), an improved version of a vanilla RNN, to obtain the contextual representation $\boldsymbol{h}_i \in \mathbb{R}^{d_h}$ for each word w_i:

$$\boldsymbol{h}_i = [\overrightarrow{\text{GRU}}(\boldsymbol{g}_i); \overleftarrow{\text{GRU}}(\boldsymbol{g}_i)] \quad (4)$$

where ";" denotes the vector concatenation.

As for the node representation in the syntax graph, the value for the representation $\boldsymbol{q}_i$ is the same as the contextual representation $\boldsymbol{h}_i$ for the word w_i, and all stop nodes share a common embedding vector which is always a zero vector because they don't carry any information for the prediction.

We get the sentence representation from the replicated random walk process, and feed it to a softmax layer to predict the sentiment polarity distribution:

$$\hat{\boldsymbol{y}} = \text{softmax}(W_o^T \boldsymbol{o} + b_o) \quad (5)$$

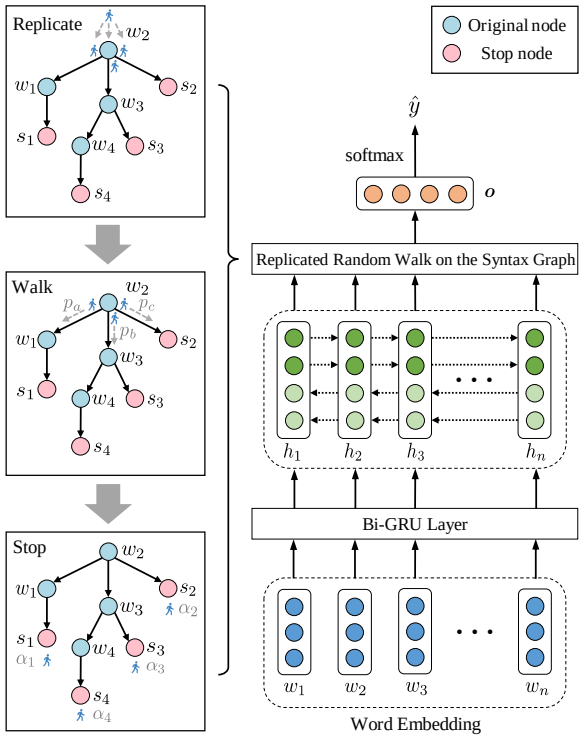


Figure 4: The overall structure of our proposed model and an illustration of the replicated random walk process.

where $\hat{\boldsymbol{y}}$ is the predicted sentiment polarity distribution, W_o and b_o are learned parameters. The overall structure of our model and an illustration of the replicated random walk process are shown in Figure 4.

Loss Function

Suppose that the training set contains K training samples $(\boldsymbol{x}_k, \boldsymbol{y}_k)$. To enforce the model to attend to a few spans that really matter for the classification, a syntactic regularization term is designed for the weight vector $\boldsymbol{\alpha}^k \in \mathbb{R}^n$ for the n contextual words in sentence $\boldsymbol{x}_k$. The weight vector $\boldsymbol{\alpha}^k$ corresponds to the weights of the words in the syntax graph. We formulate the regularization term on $\boldsymbol{\alpha}^k$ as:

$$\mathcal{L}_w = \sum_{k=1}^{K} \|\boldsymbol{\alpha}^k\|_1^2 \quad (6)$$

The final loss function consists of the cross-entropy loss, syntactic regularization term and the L_2 regularization term:

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{i=1}^{K}\sum_{j=1}^{C} y_i^j \log(\hat{y}_i^j) + \beta \mathcal{L}_w + \lambda \|\Theta\|_2^2 \quad (7)$$

where y_i^j is the ground truth sentiment polarity, C is the number of sentiment polarity categories, $\hat{y}_i^j$ denotes the predicted sentiment polarities, Θ corresponds to all of the trainable parameters, β and λ controls the influence of syntactic regularization term and L_2 regularization term.

Method	Rest14		Laptop		Twitter		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM-feature (Kiritchenko et al. 2014)	80.2#	-	70.5#	-	63.4#	63.3#	-	-
AdaRNN (Dong et al. 2014)	-	-	-	-	66.3	65.9	-	-
LSTM (Tang et al. 2016)	74.3#	63.0#	66.5#	60.1#	66.5#	64.7#	81.9*	58.1*
TD-LSTM (Tang et al. 2016)	75.6#	64.5#	68.1#	63.9#	66.6#	64.0#	82.2*	54.2*
ATAE-LSTM (Wang et al. 2016b)	77.2	-	68.7	-	-	-	83.8*	61.7*
MemNet (Tang, Qin, and Liu 2016)	81.0	-	72.2	-	-	-	83.0*	57.9*
RAM (Chen et al. 2017)	80.2	70.8	74.5	71.4	69.4	67.3	83.9*	62.1*
IAN (Ma et al. 2017)	78.6	-	72.1	-	-	-	-	-
SA-LSTM-P (Wang and Lu 2018)	81.6	-	75.1	-	69.0	-	88.7	-
PRET+MULT (He et al. 2018b)	79.1	69.7	71.2	67.5	-	-	85.6	69.8
LSTM+SynATT+TarRep (He et al. 2018a)	80.6	71.3	71.9	69.2	-	-	84.6	67.5
MGAN (Fan, Feng, and Zhao 2018)	81.3	71.9	75.4	72.5	72.5	70.8	84.4♮	63.2♮
TNet (Li et al. 2018)	80.7	71.3	76.5	71.8	75.0	73.6	86.2♮	65.2♮
HSCN (Lei et al. 2019)	77.8	70.2	76.1	72.5	69.6	66.1	-	-
MGAN (Li et al. 2019)	81.5	71.5	76.2	71.4	74.6	73.5	-	-
PWCN (Zhang, Li, and Song 2019)	81.0	72.2	76.1	72.1	-	-	-	-
TransCap (Chen and Qian 2019)	79.3	70.9	73.9	70.1	-	-	-	-
TNet-ATT(+AS) (Tang et al. 2019)	81.5	72.9	77.6	73.8	**78.6**	**77.7**	-	-
RepWalk w/o pre-trained embedding	81.8	73.2	76.2	71.9	72.4	70.4	87.7	68.7
RepWalk w/o PoS tag embedding	81.7	73.0	75.4	71.7	72.5	70.7	87.8	66.8
RepWalk w/o dependency label	80.9	71.3	75.8	71.7	71.8	69.9	87.5	64.2
RepWalk w/o syntax graph	79.2	66.1	74.1	70.0	72.1	71.0	86.9	63.0
RepWalk w/o Bi-GRU	79.3	67.6	73.2	68.3	67.8	64.4	85.0	59.4
RepWalk	**83.8**	**76.9**	**78.2**	**74.3**	74.4	72.6	**89.6**	**71.2**

Table 1: Performance comparison of different methods on the benchmark datasets. For the baseline models, the results with $*$ are retrieved from (He et al. 2018b), the results with # are retrieved from (Lei et al. 2019), the results with ♮ are produced with our implementation, other results without a symbol are retrieved from the original papers. "-" means not reported. We show the results of our model (RepWalk) in the last row, and ablated RepWalk models just above it. The best result is in **bold**.

Experiment

Datasets

We conduct experiments on four benchmark datasets, as shown below, which are from the SemEval 2014 Task 4 (Pontiki et al. 2014), Dong et al. (2014) and SemEval 2016 Task 5 (Pontiki et al. 2016). These datasets are constructed for the aspect-level sentiment analysis task.

Dataset	Positive		Negative		Neutral	
	train	test	train	test	train	test
Rest14	2164	728	807	196	637	196
Laptop	994	341	870	128	464	169
Twitter	1561	173	1560	173	3127	346
Rest16	1620	597	190	709	88	38

SemEval 2014 release two domain-specific datasets for *restaurants* (Rest14) and *laptops* (Laptop). Each training and test sample consists of a review sentence, an opinion target and the sentiment polarity towards the target. Following previous works (Chen et al. 2017), we remove samples with *conflict* polarity in the datasets. The twitter dataset is built by Dong et al. (2014), using keywords to query the Twitter API. Each tweet has a manually labeled sentiment polarity for the opinion target. The dataset of SemEval 2016 is very similar to the SemEval 2014, which is also a domain-specific dataset for *restaurants* (Rest16). We also remove the samples if the opinion target has different polarities as done in He et al. (2018a). We skip the step of the aspect-rooted dependency tree conversion for samples with no opinion target in SemEval 2016 dataset, in order to compare with the other baseline models.

Implementation Details

In our experiments, we use the pre-trained 300-dimensional GloVe vectors (Pennington, Socher, and Manning 2014) to initialize the pre-trained embeddings, and randomly initialize a 30-dimensional part-of-speech (PoS) tag embeddings. Both pre-trained and PoS tag embeddings are concatenated as word embeddings. However, we fix the pre-trained embeddings during optimization. The dimension of the Bi-GRU hidden state is set to 300. We adopt Adam (Kingma and Ba 2014) as the optimizer and follow the learning rate used in the paper. We also apply the dropout strategy (Srivastava et al. 2014) and the label smoothing technique (Szegedy et al. 2016) to alleviate overfitting. The hyperparameters β and λ are tuned for each dataset. We also make available our implementation at https://github.com/hiyouga/RepWalk.

Experimental Results

In our experiments, we compare our model (RepWalk) with a variety of baseline models as shown in Table 1. The evaluation metrics are classification accuracy and Macro-averaged

F1 score. We see that the performance of our model outperforms recent models, including attention-based models (Wang et al. 2016b; Ma et al. 2017; Wang and Lu 2018; Fan, Feng, and Zhao 2018), memory-based models (Tang, Qin, and Liu 2016; Chen et al. 2017), and those which use syntactic information (Dong et al. 2014; He et al. 2018a; Zhang, Li, and Song 2019), in both accuracy and Macro-F1. However, we see that our model cannot outperform TNet (Li et al. 2018) on the Twitter dataset. As seen clearly from our model structure and our ablation studies, the performance of our model is largely attributed to the information extracted from the dependency tree. However, reviews in the Twitter dataset are usually short and largely informal. Hence, the syntactic tree may not always hold bringing about noise in the dataset. We perform an analysis on the relationship between the model performance and the sentence quality to study this further.

Ablation Study

We perform an ablation study on the benchmark datasets to investigate the relevance of each component on model performance. In particular, we observe the contribution of syntactic information on model performance. The results are shown in Table 1.

In one model ablation, we remove the pre-trained embeddings and only use the PoS tag embeddings as the node representations to compute the probabilities on the edges of the syntax graph. We refer to this ablated model as "RepWalk w/o pre-trained embedding" in the table. RepWalk outperforms RepWalk w/o pre-trained embedding by leveraging on contextual information from the pre-trained embeddings. Nevertheless, we see that RepWalk w/o pre-trained embedding outperforms LSTM+SynATT+TarRep (He et al. 2018a) which takes into account both contextual and syntactic information. RepWalk w/o pre-trained embedding also shows competitive performance with other strong baselines. The results suggest that our approach for integrating syntactic structure is very effective and that the pre-trained embeddings can bring about improvement.

In a similar ablation study, we simply remove the PoS tag embeddings in the node representation of RepWalk. We refer to this ablated model as "RepWalk w/o PoS tag embedding" in the table. The results of the RepWalk w/o PoS tag embedding demonstrate that both pre-trained embeddings and its associated PoS tag embeddings complement each other to enrich the representation for the classification task.

We study the importance of the edge type in the syntax graph. For this purpose, we perform an ablation study where we remove all labels on the edges in the syntax graph. We refer to this ablated model as "RepWalk w/o dependency label". We observe that there is a significant drop in the performance of RepWalk. The results suggest that the dependency label is an essential component in the dependency tree. Methods such as He et al.; Zhang, Li, and Song (2018a; 2019) ignore the dependency label information and therefore cannot achieve a better performance relative to the performance of RepWalk.

To study the overall improvement brought by the syntactic information to the model, we remove the syntax graph and use a mean-pool function to distill a final embedding from the Bi-GRU layer. We refer to such an ablated model as "RepWalk w/o syntax graph". The results of the RepWalk w/o syntax graph are incomparable with the ablated models mentioned so far, except the model performance on the Twitter dataset. We can see that using syntactic information cannot make much improvement if this feature contains lots of noise. As for the Bi-GRU layer, when we remove this component, the results of the RepWalk w/o Bi-GRU show that the Bi-GRU network plays a key role in getting contextual representation for this task.

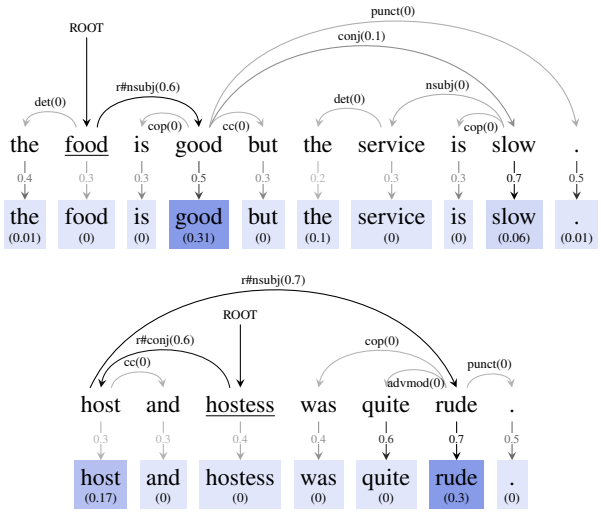


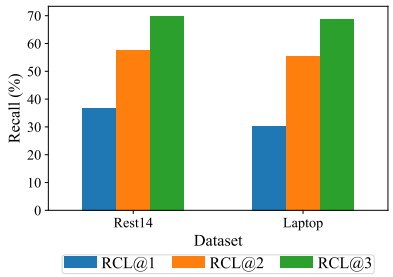
Figure 5: The underlined word denotes the aspect term. Darker edges denote paths activated with high probability. A word with a relatively darker shade indicates a word with a large weight, and has a higher probability of being highlighted in the syntax graph.

Case Study

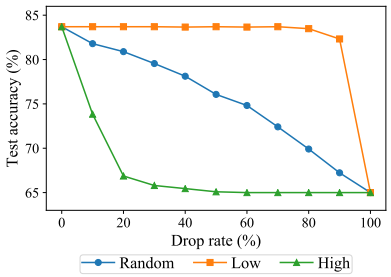
We visualize the syntax graph with probability values on the edges and words of 2 samples shown in Figure 5. We find that our model is able to capture informative contextual words in various cases. Especially, it highlights these words by activating the edges in the syntax graph.

As we can see from the first example, the path connecting the adjective "good" and the target aspect "food" receives a higher probability to be activated relative to other paths in the syntax graph. Thus, RepWalk assigns a high weight on the word "good", which is the informative word expressing sentiments on "food". We also see that the sentiments expressed on other aspects do not affect the current prediction.

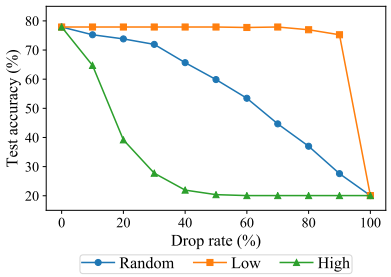
Obviously, the informative contextual word and the aspect term are not always directly connected in the syntax graph. The second example shows that the model can successfully capture the informative word concerning the aspect term. From these examples, we can observe that the proposed model is capable of capturing the important contextual information with the help of syntactic structure so as to perform an aspect-level sentiment classification.



(a) The recall value on the Rest14 and Laptop dataset.



(b) The accuracy curve under different drop rates on the Rest14 dataset.



(c) The accuracy curve under different drop rates on the Laptop dataset.

Figure 6: Analysis of the experimental results for understanding the model behaviour.

Understanding Model Behaviour

The replicated random walk computes a weight α_i for the word w_i which indicates the importance of the contextual feature of this word. Thus our model can focus on the expressive words in the context. To prove the effectiveness, we collect pairs of weights and words from the pre-trained model on Rest14 and Laptop datasets. To evaluate whether the words play a greater role in the classification task, we compare the words which have been assigned high weights in the sentence with the annotated opinion words provided by Wang et al. (2016a). For this purpose, we adopt the "recall" metric to measure how many opinion words are covered by the top-1, top-2 or top-3 words with high weights. The results are shown in Figure 6(a) as "RCL@1", "RCL@2" and "RCL@3" for the respective top-1, top-2 and top-3 words. From the results, we find that lots of opinion words are covered by the contextual words focused by RepWalk. The results suggest that RepWalk can properly capture informative contextual information which are expressive in the sentence concerning the aspect term.

We conduct another well-designed experiment to validate our hypothesis by investigating the relationship between contextual words with different levels of importance and the classifier's performance, where contextual words are represented by their features. We first rank all contextual words based on their weights in descending order, and evaluate the performance of the classifier in three settings. In the first setting, we randomly drop the words for different proportions. In the second setting, we drop the words from the bottom of the ranked list for different proportions. In the third setting, we drop the words from the top of the ranked list for different proportions. The performance is shown by "Random", "Low" and "High" curves in Figure 6(b,c) respectively. Experimental results show the classifier's performance indeed highly depends on the contextual features of highlighted words in the sentence.

Model Performance vs. Sentence Quality

We perform a systematic study on the unsatisfactory performance of RepWalk on the Twitter dataset. Based on our analysis, we find that errors can be broadly attributed to the incorrect parsing tree result due to the ungrammatical sentences in online posts. Similarly to Zhang et al. (2018), we use the LanguageTool[1] to judge the grammatical correctness of the sentences. More specifically, we use the number of errors in spelling and grammar in the sentences to measure sentence correctness. Figure 7 shows the relative performance of our model compared with TNet (Li et al. 2018) and the grammatical score of the sentences on four benchmark datasets. Results suggest that sentence with incorrect spelling and grammar will hurt the model's performance.

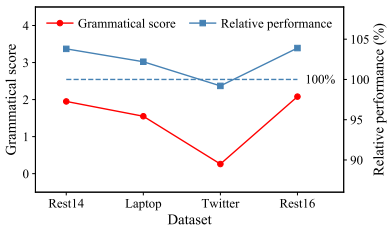


Figure 7: The relative performance of our model compared with TNet (Li et al. 2018) and the grammatical score of the sentences on four benchmark datasets.

Conclusion

In this paper, we propose a novel neural network model (RepWalk) which effectively leverages syntactic structures to improve sentence representations. Based on a replicated random walk process, we show that our model is able to successfully capture informative contextual features of the sentence. We conduct experiments on benchmark datasets, showing that our proposed method performs better than the very recent state-of-the-art methods on most of the benchmark datasets for ALSC task. More importantly, we realized that the performance of the model is hinged on the ability to parse sentences into the correct dependency tree, where these sentences are grammatically correct and free from spelling errors. However, we find that reviews are largely short and informal bringing about a bottleneck when parsing into correct syntactic dependencies. This is noted on the performance of RepWalk on the Twitter dataset. In the future, we will explore approaches of dealing with sentences which are largely informal or short for ALSC.

[1] https://languagetool.org/

Acknowledgments

This work is supported partly 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), by the Beijing S&T Committee (No. Z191100008619007), and by the Fundamental Research Funds for the Central Universities.

References

Chen, D., and Manning, C. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, 740–750.

Chen, Z., and Qian, T. 2019. Transfer capsule network for aspect level sentiment classification. In *ACL*, 547–556.

Chen, P.; Sun, Z.; Bing, L.; and Yang, W. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of EMNLP*, 452–461.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; and Xu, K. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of ACL*, 49–54.

Fan, F.; Feng, Y.; and Zhao, D. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of EMNLP*, 3433–3442.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 2018a. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of COLING*, 1121–1131.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 2018b. Exploiting document knowledge for aspect-level sentiment classification. In *Proceedings of ACL*, 579–585.

Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; and Zhao, T. 2011. Target-dependent twitter sentiment classification. In *Proceedings of ACL*, 151–160.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiritchenko, S.; Zhu, X.; Cherry, C.; and Mohammad, S. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of SemEval*, 437–442.

Lei, Z.; Yang, Y.; Yang, M.; Zhao, W.; Guo, J.; and Liu, Y. 2019. A human-like semantic cognition network for aspect-level sentiment classification. In *AAAI*, 6650–6657.

Li, X.; Bing, L.; Lam, W.; and Shi, B. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of ACL*, 946–956.

Li, Z.; Wei, Y.; Zhang, Y.; Zhang, X.; and Li, X. 2019. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *Proceedings of AAAI*, 4253–4260.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

Ma, D.; Li, S.; Zhang, X.; and Wang, H. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of IJCAI*, 4068–4074.

Pang, B.; Lee, L.; et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 1532–1543.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, 27–35.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.-S.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval*, 19–30.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of CVPR*, 2818–2826.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING*, 3298–3307.

Tang, J.; Lu, Z.; Su, J.; Ge, Y.; Song, L.; Sun, L.; and Luo, J. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *ACL*, 557–566.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP*, 214–224.

Wang, B., and Lu, W. 2018. Learning latent opinions for aspect-level sentiment classification. In *AAAI*.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of EMNLP*, 616–626.

Wang, Y.; Huang, M.; Zhao, L.; et al. 2016b. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of EMNLP*, 606–615.

Wang, S.; Mazumder, S.; Liu, B.; Zhou, M.; and Chang, Y. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of ACL*, 957–967.

Xue, W., and Li, T. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of ACL*, 2514–2523.

Zhang, R.; Hu, Z.; Guo, H.; and Mao, Y. 2018. Syntax encoding with application in authorship attribution. In *Proceedings of EMNLP*, 2742–2753.

Zhang, C.; Li, Q.; and Song, D. 2019. Syntax-aware aspect-level sentiment classification with proximity-weighted convolution network. In *Proceedings of SIGIR*, 1145–1148.