# Improved Review Sentiment Analysis with a Syntax-Aware Encoder

Jiangfeng Zeng[1], Ming Yang[2], Ke Zhou[1(✉)], Xiao Ma[3], Yangtao Wang[1], Xiaodong Xu[1], and Zhili Xiao[4]

[1] Huazhong University of Science and Technology, Wuhan, China
{jfzeng,k.zhou,ytwbruce,xiaodong-xu}@hust.edu.cn
[2] Wuhan cciisoft Co., Ltd., Wuhan, China
mingyangIris@gmail.com
[3] Zhongnan University of Economics and Law, Wuhan, China
cindyma@zuel.edu.cn
[4] Tencent Inc., Shenzhen, China
tomxiao@tencent.com

**Abstract.** Review sentiment analysis has drawn a lot of active research interest because of the explosive growth in the amount of available reviews in our day-to-day activities. The current review sentiment classification work often models each sentence as a sequence of words, thus simply training sequence-structured recurrent neural networks (RNNs) end-to-end and optimizing via stochastic gradient descent (SGD). However, such sequence-structured architectures overlook the syntactic hierarchy among the words in a sentence. As a result, they fail to capture the syntactic properties that would naturally combine words to phrases. In this paper, we propose to model each sentence of a review with an attention-based dependency tree-LSTM, where a sentence embedding is obtained relying on the dependency tree of the sentence as well as the attention mechanism in the tree structure. Then, we feed all the sentence representations into a sequence-structured long short-term memory network (LSTM) and exploit attention mechanism to generate the review embedding for final sentiment classification. We evaluate our attention-based tree-LSTM model on three public datasets, and experimental results turn out that it outperforms the state-of-the-art baselines.

**Keywords:** Sentiment analysis · Recurrent neural networks · tree-LSTM · Syntax-aware

## 1 Introduction

As a branch of text-based multimedia analysis, sentiment analysis [20] is a challenging study of vital importance in natural language processing (NLP). Recently, more and more attention has been focused on document-level sentiment analysis in the research community these years because of two main reasons: (1) over the past several decades, there has been an explosive growth in

J. Shao et al. (Eds.): APWeb-WAIM 2019, LNCS 11642, pp. 73–87, 2019.
https://doi.org/10.1007/978-3-030-26075-0_6

the amount of reviews from social networks like Twitter, Facebook, Instagram, etc; (2) successfully classifying the review sentiment is crucial to understanding customer preferences and enabling new businesses such as customized recommendation [7].

Existing approaches can be divided into two classes: traditional machine learning models and neural network models. Traditional machine learning methods are dedicated to manually engineering an abundance of useful features like bag-of-words [20], sentiment lexicons [21,23] and social networks [6] to build the classification models. Although traditional methods like Support Vector Machine (SVM) [19] obtained a good performance on this task, they are always blamed for complicated and labor-intensive feature engineering. Since deep learning techniques have been successfully applied to many tasks in both computer vision and natural language processing, some recent studies start to address sentiment classification using well-designed neural networks. Compared with feature based traditional machine learning models, neural networks models have achieved promising results on sentiment analysis for their capability to learn powerful and semantic feature representations from original data without careful handcraft feature engineering [8,11,16,25,26]. Document-level sentiment analysis is a challenging task and far from being solved. Without considering the document structure, cached long short-term memory networks [31] devises a cache mechanism to divide memory into several groups via different forgetting rates, thus enabling the network to capture the overall sentiment information better within a recurrent unit. [2] and [27] realize the vital importance of document hierarchy structure and build hierarchical models to deal with document-level sentiment analysis. To further improve the classification accuracy, attention mechanism is exploited to select important word-level and sentence-level features hierarchically [5,28,33]. In [28], user(product) matrix representation is not well defined because of the data insufficiency in terms of those users(products) with limited reviews. Chen et al. [5] develop NSC+UPA that achieves the state-of-the-art. However, word vectors didn't combine neither user nor product information before attention weights are computed so that the sentence representation misses the opportunity to enrich the sentence semantics with user and product information.

Although these attention based sequence-structured RNNs have shown to significantly improve the sentiment classification results, several prominent researchers push back against this cognition that language is just sequences of words [4,22,34]. In other words, linguistic structure is coming back. Consequently, in this paper, we model each sentence of a review with an attention-based dependency tree-LSTM, where a sentence embedding is obtained relied on the dependency tree of the sentence as well as the attention mechanism in the tree structure. Then, all the sentence representations are fed into a sequence-structured long short-term memory network (LSTM) and the attention mechanism is utilized to generate the review embedding for final sentiment classification. The major contributions of this paper are three-fold:

- We propose an attention-based dependency tree-LSTM to model each sentence, which computes attention based on the dependency parsing tree of a sentence.
- Because of the high computation complexity and the big memory capacity, current tree-LSTM models don't run in batch mode. We implement the attention based tree-LSTM in batch mode, which greatly reduces the training time.
- Experimental results on three open datasets are conducted to demonstrate the effectiveness of our tree-structured model for review sentiment analysis.

The remainder of this paper is organized as follows. First, we present the proposed hierarchical architecture in Sect. 2. Then, Sect. 3 displays extensive experiments conducted to demonstrate the superiority of the proposed architecture. Afterwards, in Sect. 4, we discuss the related works on sentiment analysis. Finally, we draw a conclusion and envision the future in Sect. 5.

## 2 The Proposed Approach

Let us first formulate the problem of review sentiment classification.We take as input a set of training reviews $D = \{(d_1, y_1), (d_2, y_2), ..., (d_n, y_n)\}$, where $d_i$ is a document-level comment containing more than one sentence and $y_i \in Y$ depicts the sentiment label (e.g., $Y = \{1, 2, 3, 4, 5\}$ means that the rating value is ranged from "one star" to "five star"). Thus, document-level sentiment classification aims at inferring the sentiment label of a review.

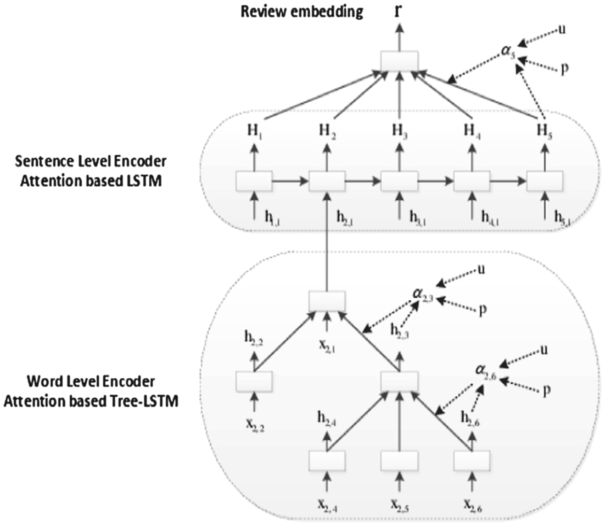

**Fig. 1.** Overview of the proposed hierarchical architecture for review sentiment analysis.

Considering the hierarchical document structure, we model each review hierarchically as done in [5,33]: word level (from word to sentence) and sentence level (from sentence to document). In word level, we devise an attention-based dependency tree-LSTM to encode each sentence of a review. In sentence level, a sequence-structured LSTM encoder followed by an attention layer is developed to generate attentive representations for reviews. Note that the attention weights in both word level and sentence level are computed by using global user preferences and product characteristics as guidance. A high-level illustration of our proposed approach is shown in Fig. 1. For simplicity and generality, we suppose that the root of each dependency tree is the first word of each sentence. We then examine each module detailedly and give intuitions about its formulation.

### 2.1 Embedding from Word to Sentence

For each sentence of a review, the sentence representation is obtained by modeling the words belonging to the sentence. Suppose that the $j$-th sentence $s_j$ contains $n_j$ words, and is denoted as $s_j = \{w_{j,1}, w_{j,2}, ..., w_{j,n_j}\}$, where $w_{j,k} \in R^{K_1}$ is from the pre-trained word embeddings by *word2vec* [17]. Draw inspirations from the aspect embedding devised by [30], we vectorize user preferences and product characteristics as user embedding $u \in R^{L_1}$ for each user and product embedding $p \in R^{L_2}$ for each product, respectively. Note that user embedding and product embedding are treated as training parameters like other model parameters. In addition, we concatenate each word vector $w_{j,k}$ with user vector $u$ and product vector $p$ as inputs for our attention-based tree-LSTM. For simplicity in notation, we denote $x_{j,k} = [w_{j,k}; u; p]$. It is worth mentioning that such a concatenation design contributes a lot to enhancing the sentence semantics with user and product features. However, the previous methods neglect this design. For example, Tang et al. [28] represent each user (product) with a matrix which is multiplied with each word embedding $w_{j,k}$ to get the inputs for sentence embeddings. In [5], neither user and product information is used as inputs to encode sentences.

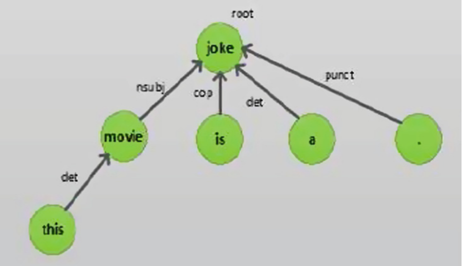

**Fig. 2.** An example dependency tree parsed by the Stanford Neural Network Dependency Parser.

Evidently, sequence-structured RNNs are increasingly incapable of capturing long-range dependencies. To make full advantage of the syntactic structure in each sentence, we employ a tree-LSTM [26] to generate a hidden vector for each word. In detail, we first parse each sentence into a dependency tree or a constituency tree using the Stanford Neural Network Dependency Parser [3]. In this work, we choose the $child - sum$ schema [26] for tree-LSTM to model each sentence. An example dependency tree for the sentence "This movie is a joke." is illustrated in Fig. 2. We can see that an dependency tree is composed of nodes corresponding to words in the sentence and edges representing the syntax relationships between parent node and its child nodes. Note that arrow directions reveal the computing order, that is to say, parent node is computed only after its child nodes have been computed. Then the representation for the root node of the tree is regarded as the sentence representation.

Given a dependency tree of a sentence, we denote $C_{j,k}$ as the set of children for word $w_{j,k}$. Formally, for each input vector $x_{j,k}$, we output a vector $h_{j,k} \in R^{K_2}$, by computing a series of neuron activations for an input gate $g_{j,k}^{(i)}$, several forget gates $g_{j,k,l}^{(f)}$, a memory cell state $g_{j,k}^{(c)}$ and an output gate $g_{j,k}^{(o)}$:

$$\overline{h}_{j,k} = \sum_{w_{j,l} \in C_{j,k}} h_{j,l} \tag{1}$$

$$g_{j,k}^{(i)} = \sigma(W^{(i)} x_{j,k} + U^{(i)} \overline{h}_{j,k} + b^{(i)}) \tag{2}$$

$$g_{j,k,l}^{(f)} = \sigma(W^{(f)} x_{j,k} + U^{(f)} h_{j,l} + b^{(f)}) \tag{3}$$

$$\overline{g}_{j,k}^{(c)} = tanh(W^{(c)} x_{j,k} + U^{(c)} \overline{h}_{j,k} + b^{(c)}) \tag{4}$$

$$g_{j,k}^{(c)} = g_{j,k}^{(i)} \odot \overline{g}_{j,k}^{(c)} + \sum_{w_{j,l} \in C_{j,k}} g_{j,k,l}^{(f)} \odot g_{j,l}^{(c)} \tag{5}$$

$$g_{j,k}^{(o)} = \sigma(W^{(o)} x_{j,k} + U^{(o)} \overline{h}_{j,k} + b^{(o)}) \tag{6}$$

$$h_{j,k} = g_{j,k}^{(o)} \odot tanh(g_{j,k}^{(c)}) \tag{7}$$

where $\odot$ is an element-wise product, and $\Theta^{(tree-lstm)} = \{W^{(i)} \in R^{K_2 \times K_1}, U^{(i)} \in R^{K_2 \times K_2}, b^{(i)} \in R^{K_2}, W^{(f)} \in R^{K_2 \times K_1}, U^{(f)} \in R^{K_2 \times K_2}, b^{(f)} \in R^{K_2}, W^{(c)} \in R^{K_2 \times K_1}, U^{(c)} \in R^{K_2 \times K_2}, b^{(c)} \in R^{K_2}, W^{(o)} \in R^{K_2 \times K_1}, U^{(o)} \in R^{K_2 \times K_2}, b^{(o)} \in R^{K_2}\}$ is the set of parameters for tree-LSTM.

We introduce attention to tree-LSTM based on the dependency tree. In particular, we add attention into both Eqs. 1 and 5. In terms of Eq. 1, to sum up the children hidden vectors is arbitrary. Therefore, we define

$$\begin{aligned} v_{j,l}^{(s)} &= h_{j,l}^T W_2^{(s)} u + h_{j,l}^T W_3^{(s)} p \\ \beta_{j,l}^{(s)} &= \eta^{(s)} \cdot f(W_1^{(s)} h_{j,l} + b^{(s)}) + v_{j,l}^{(s)} \\ \alpha_{j,l} &= \frac{\exp(\beta_{j,l}^{(s)})}{\sum_{w_{j,i} \in C_{j,k}} \exp(\beta_{j,i}^{(s)})} \end{aligned} \tag{8}$$

where $v_{j,l}^{(s)}$ is a term introduced to indicate the relevance between word hidden vector $h_{j,l}$ and user embedding $u$, as well as product embedding $p$. $f(\cdot)$ is a nonlinear function like sigmoid or tanh. $\Theta^{(Att_1)} = \{W_1^{(s)} \in R^{K_2 \times K_2}, W_2^{(s)} \in R^{K_2 \times L_1}, W_3^{(s)} \in R^{K_2 \times L_2}, b^{(s)} \in R^{K_2}, \eta^{(s)} \in R^{K_2}\}$ is the attention network parameters to be learned. Then, by aggregating all the children hidden vectors according to $\alpha_{j,l}$, Eq. 1 can be updated as

$$\overline{h}_{j,k} = \sum_{w_{j,l} \in C_{j,k}} \alpha_{j,l} h_{j,l} \tag{9}$$

Similarly, for Eq. 5, we compute attention weights as

$$\begin{aligned} \overline{g}_{j,k,l}^{(c)} &= g_{j,k,l}^{(f)} \odot g_{j,l}^{(c)} \\ v_{j,k,l}^{(t)} &= (\overline{g}_{j,k,l}^{(c)})^T W_2^{(t)} u + (\overline{g}_{j,k,l}^{(c)})^T W_3^{(t)} p \\ \beta_{j,k,l}^{(t)} &= \eta^{(c)} \cdot f(W_1^{(t)} \overline{g}_{j,k,l}^{(c)} + b^{(t)}) + v_{j,k,l}^{(t)} \\ \alpha_{j,k,l} &= \frac{\exp(\beta_{j,k,l}^{(t)})}{\sum_{w_{j,i} \in C_{j,k}} \exp(\beta_{j,k,i}^{(t)})} \end{aligned} \tag{10}$$

where $\Theta^{(Att_2)} = \{W_1^{(t)} \in R^{K_2 \times K_2}, W_2^{(t)} \in R^{K_2 \times L_1}, W_3^{(t)} \in R^{K_2 \times L_2}, b^{(t)} \in R^{K_2}, \eta^{(t)} \in R^{K_2}\}$ is the attention network parameters to be learned. Thus Eq. 5 can be updated as

$$g_{j,k}^{(c)} = g_{j,k}^{(i)} \odot \overline{g}_{j,k}^{(c)} + \sum_{w_{j,l} \in C_{j,k}} \alpha_{j,k,l} \overline{g}_{j,k,l}^{(c)} \tag{11}$$

Finally, we take the output hidden vector of the root of the dependency tree as the ultimate sentence embedding.

### 2.2 Embedding from Sentence to Document

As shown in Fig. 1, we regard each review as a sequence of sentences. Given the sentence embeddings of a review generated in word level, we develop a sequence-structured LSTM [10], where $\Theta^{lstm} = \{W_{lstm}^{(i)} \in R^{K_2 \times K_2}, U_{lstm}^{(i)} \in R^{K_2 \times K_2}, b_{lstm}^{(i)} \in R^{K_2}, W_{lstm}^{(f)} \in R^{K_2 \times K_2}, U_{lstm}^{(f)} \in R^{K_2 \times K_2}, b_{lstm}^{(f)} \in R^{K_2}, W_{lstm}^{(c)} \in R^{K_2 \times K_2}, U_{lstm}^{(c)} \in R^{K_2 \times K_2}, b_{lstm}^{(c)} \in R^{K_2}, W_{lstm}^{(o)} \in R^{K_2 \times K_2}, U_{lstm}^{(o)} \in R^{K_2 \times K_2}, b_{lstm}^{(o)} \in R^{K_2}\}$. For simplicity in notation, we denote the outputs of LSTM for sentences $\{s_1, s_2, ..., s_m\}$ as $\{H_1, H_2, ..., H_m\}$. Based on this intuition that not all the sentences contribute equally to the semantic meaning of a review, we also exploit attention mechanism to select sentences of significant importance.

Attention weights are computed as follows:

$$\begin{aligned} v_j^{(r)} &= H_j^T W_2^{(r)} u + h_j^T W_3^{(r)} p \\ \beta_j^{(r)} &= \eta^{(r)} \cdot f(W_1^{(r)} H_j + b^{(r)}) + v_j^{(r)} \\ \alpha_j &= \frac{\exp(\beta_j^{(r)})}{\sum_{i=1}^{m} \exp(\beta_j^{(r)})} \end{aligned} \tag{12}$$

where $\Theta^{(Att_3)} = \{W_1^{(r)} \in R^{K_2 \times K_2}, W_2^{(r)} \in R^{K_2 \times L_1}, W_3^{(r)} \in R^{K_2 \times L_2}, b^{(r)} \in R^{K_2}, \eta^{(r)} \in R^{K_2}\}$ is defined as the attention network parameters. Then we aggregate all the sentence hidden vectors according to attention weights $\alpha_j$ and the review embedding is computed by

$$r = \sum_{j=1}^{m} \alpha_j H_j \tag{13}$$

### 2.3 Sentiment Classification

Since review embeddings are hierarchically extracted from the words and sentences, they are high level semantic representations for reviews. Hence, we use them to train our sentiment classifier. We first use a nonlinear layer to project review embedding $r$ into the target space of $C$ classes:

$$\hat{r} = tanh(W_r r + b_r) \tag{14}$$

where $C$ is the number of sentiment classes, and $\Theta^{(classifier)} = \{W_r \in R^{K_2 \times C}, b_r \in R^C\}$ is the parameters to be learned. Afterwards, a softmax layer is adopted to compute the sentiment distribution:

$$p_c = \frac{\exp(\hat{r}_c)}{\sum_{z=1}^{C} \exp(\hat{r}_z)} \tag{15}$$

### 2.4 Model Training

In our work, we need to optimize all the parameters notated as $\Theta = \{\Theta^{(tree-lstm)}, \Theta^{lstm}, \Theta^{(Att_1)}, \Theta^{(Att_2)}, \Theta^{(Att_3)}, \Theta^{(classifier)}, u, p\}$. Cross entropy with $L_2$ regularization is defined as the loss function for optimization when training:

$$L = -\sum_{d \in D} \sum_{c=1}^{C} y_c(d) \cdot \log(p_c(d)) + \lambda L_2(\Theta) \tag{16}$$

where $y_c(d)$ is the golden sentiment distribution and $\lambda$ means the coefficient for $L_2$ regularization.

# 3 Evaluation

In this section, we present our experiment settings and conduct experiments on the task of document-level review sentiment analysis.

## 3.1 Experimental Settings

We evaluate the proposed approach on three real-world datasets, i.e., Yelp13, Yelp14 and IMDB, all of which are from [28]. Each record in the datasets is composed of a user ID, a product ID, a review and a rating. We summarize the statistics of our used datasets in Table 1. The Yelp Dataset Challenge produces Yelp 2013 and Yelp 2014 which contain a large number of restaurant reviews labeled with stars ranging from 1 to 5. IMDB contains 84919 movie reviews labeled with stars ranging from 1 to 10. We initialize the word vectors with 200-dimensional *word2vec* [17], and initialize user embedding and product embedding with a 200-dimensional zero vector. During training, word vectors, user embedding and product embedding are all fine-tuned as parameters. We set batch size to be 32, i.e., 32 documents, set $L_2$-regularization weight to be 0.00001 and initialize learning rate to be 0.05 for AdaDelta.

**Table 1.** Dataset description

| Datasets | #docs | #users | #products | #docs/ user | #docs/ product | #sens/doc | #words/sen |
|---|---|---|---|---|---|---|---|
| *IMDB* | 84,919 | 1,310 | 1,635 | 64.82 | 51.94 | 16.08 | 24.54 |
| *Yelp13* | 78,966 | 1,631 | 1,633 | 48.42 | 48.36 | 10.89 | 17.38 |
| *Yelp14* | 231,163 | 4,818 | 4,194 | 47.97 | 55.11 | 11.41 | 17.26 |

## 3.2 Evaluation Metrics

Two metrics are utilized to evaluate our model. *Accuracy* measures the overall sentiment classification performance, is formalized as:

$$Accuracy = \frac{T}{N} \tag{17}$$

where $T$ is the number of reviews correctly predicted and $N$ is the size of testing dataset. Another metric is $RMSE$, which calculates the divergences between predicted labels and ground truth labels and can be computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (gd_i - pr_i)^2}{N}} \tag{18}$$

where $gd_i$ and $pr_i$ are golden truth label and predicted label respectively.

### 3.3 Baselines

We list several baseline methods for comparisons with our method as follows.

**Majority** infers the sentiment category of the test dataset according to the majority sentiment category in training dataset.

**Trigram** trains a Support Vector Machine with n-gram features, i.e., unigrams, bigrams and trigrams.

**TextFeature** is an another SVM-based method which is trained using word and character n-grams, sentiment lexicon features, etc.

**AvgWordvec** builds 200-dimensional word vectors using *word2vec* [17] and averages all the word vectors of each review as final review representation to train a SVM classifier.

**SSWE** uses sentiment-specific word embeddings generated by [29] to train a SVM classifier.

**RNTN+RNN** makes use of RNTN [25] to generate sentence embeddings which then are processed by a RNN to produce review representations for final classification.

**Paragraph Vector** unsupervisedly learns representations for sentences and documents [12].

**JMARS** is proposed in recommender systems [7]. It combines user information and aspects of a review with collaborative filtering and topic modeling.

**UPNN** is first to take user and product information into consideration when addressing review sentiment classification. Tang et al. [28] devises user matrix and product matrix which are concatenated with document representation for final sentiment classification.

**NSC** & **NSC+LA** & **NSC+UPA** are developed by [5], core of which is a sequence-structured LSTM. **NSC** encodes words and sentences of one review in a hierarchical manner, but ignores user and product information. **NSC+LA** use the local attention without using user and product information. While **NSC+UPA** introduces global user preferences and product characteristics as attention guiders over different semantic levels of a document, therefore generating robust and semantic document representations.

### 3.4 Model Comparisons

We conduct our comparison experiments in two scenarios: with user and product information, denoted as “with U and P”, and otherwise denoted as “no U, no P”. The classification accuracy and $RMSE$ results are shown in Table 2.

From the rows noted by “no U, no P” in Table 2, we can see that our proposed method outperforms almost all of the baselines, which indicates that linguistic structure based neural networks have advantage over sequence-structured neural networks when addressing document-level review sentiment analysis. From the rows noted by “with U and P”, we can see that global user preferences and product characteristics play an important role and attention mechanism successfully captures useful semantics related to the user and product which contribute to better training the final sentiment classifier. Evidently, no matter modeling

**Table 2.** Sentiment classification results of our model against competitor models on IMDB, Yelp13 and Yelp14. Acc(Accuracy) and RMSE are the two used evaluation metrics. Best results in each group are in bold.

| Settings | Methods | IMDB | | Yelp13 | | Yelp14 | |
|---|---|---|---|---|---|---|---|
| | | Acc | RMSE | Acc | RMSE | Acc | RMSE |
| no U, no P | Majority | 0.196 | 2.495 | 0.411 | 1.060 | 0.392 | 1.097 |
| | Trigram | 0.399 | 1.783 | 0.569 | 0.814 | 0.577 | 0.804 |
| | TextFeature | 0.402 | 1.793 | 0.556 | 0.845 | 0.572 | 0.800 |
| | AvgWordvec+SVM | 0.304 | 1.985 | 0.526 | 0.898 | 0.530 | 0.893 |
| | SSWE+SVM | 0.312 | 1.973 | 0.549 | 0.849 | 0.557 | 0.851 |
| | Paragraph Vector | 0.341 | 1.814 | 0.554 | 0.832 | 0.564 | 0.802 |
| | RNTN+RNN | 0.400 | 1.764 | 0.574 | 0.804 | 0.582 | 0.821 |
| | UPNN | 0.405 | 1.629 | 0.577 | 0.812 | 0.585 | 0.808 |
| | NSC | 0.443 | 1.465 | 0.627 | 0.701 | **0.637** | **0.686** |
| | NSC+LA | 0.487 | 1.381 | 0.631 | 0.706 | 0.630 | 0.715 |
| | **Ours** | **0.493** | **1.378** | **0.635** | **0.700** | 0.634 | 0.689 |
| with U and P | Trigram+UPF | 0.404 | 1.764 | 0.570 | 0.803 | 0.576 | 0.789 |
| | TextFeature+UPF | 0.402 | 1.774 | 0.561 | 0.822 | 0.579 | 0.791 |
| | JMARS | N/A | 1.773 | N/A | 0.985 | N/A | 0.999 |
| | UPNN | 0.435 | 1.602 | 0.596 | 0.784 | 0.608 | 0.764 |
| | NSC+UPA | 0.533 | 1.281 | **0.650** | **0.692** | 0.667 | 0.654 |
| | **Ours+UPA** | **0.538** | **1.276** | 0.649 | 0.697 | **0.669** | **0.650** |

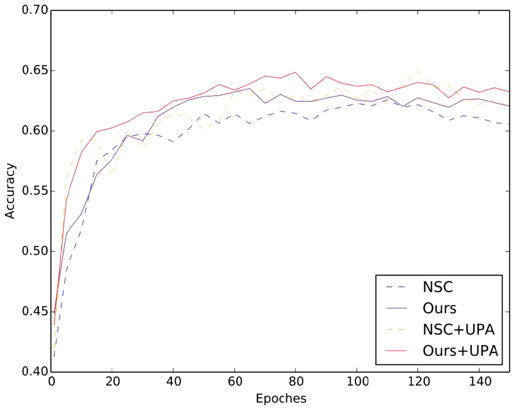

**Fig. 3.** Convergence speed experiment on Yelp13.

with or without user and product information, our approach is demonstrated to achieve consistent improvements compared with other competitors.

In order to clearly display the advantages of our syntax-aware method over other competitive baselines, we conduct the convergence speed experiment on

Yelp13, results of which are shown in Fig. 3. X-axis is the iteration epochs and Y-axis is the classification accuracy predicted on validation dataset. Note that one epoch in our experiments does not mean to run out the training dataset. Since the datasets used here are too large that We train 3200 random selected reviews one epoch and then validate on our validation dataset, which is also used in the work of [5]. As representatives of the current state-of-the-art sequence-structured methods, the comparison is made between our syntax-aware method and [5]. From Fig. 3, it can be observed that in terms of convergence rate our syntax-aware method beats [5]. The reason why our approach converges faster is that syntactic linguistic features are integrated into review representations and efficiently guides the classifier.

When comparing with other baselines, we have two instructive findings. First, to concatenate each word embedding with user embedding and product embedding as the inputs for neural network models contributes a lot to enriching the sentence semantics with user and product information, which increases the exposure rate of important semantics related to the user and product when computing attention weights. Second, linguistic structure based RNNs are in theory superior to sequence-structured RNNs. But experiment results show that our model based on dependency tree achieves slightly better results than the model proposed in [5]. From the implementation of tree-LSTM, we find that tree-LSTM suffers from the underfitting issue: the same shared compositional function throughout the whole compositional procedure results in the lack of expressive power.

## 4 Related Works

In essence, sentiment analysis can be thought of as a special kind of text classification. It is obvious that the more effective the extracted features are, the better performance the text classifiers will obtain. Earlier researches on sentiment analysis mostly focus on designing useful handcraft features, which is time-consuming and demands for expert knowledge. For example, Pang et al. [20] exploit machine learning algorithms to train classifiers with bag-of-words. Sentiment lexicons [21,23] are also utilized to improve the classification performance. Cheng et al. [6] mine useful features from social networks. Over the past several years, deep neural networks win a high reputation in substantial applications for automatic representation learning [35]. [11] and [37] represent sentences by using convolutional neural networks (CNNs) to model sentences like images. Recurrent neural networks (RNNs) emerge as methods dedicated to processing sequential data, and have achieved great success in a large number of NLP tasks. Introducing two kinds of syntactic parsing trees, Tai et al. [26] devise tree-structured long-short term memory networks (TreeLSTMs) which work better compared with sequential LSTMs. In the work of Xu et al. [31], cached long short-term memory networks are proposed to solve document-level sentiment classification. Taking into consideration the hierarchical structure of documents, some researchers attempt to model documents hierarchically [2,27]. It has been proved again and again that attention mechanism is beneficial to selecting valuable information and getting rid of useless information for substantial applications including image generation [9,18], machine translation [1,14], image caption [32], natural language inference [24], deep hashing [13,38], etc. Attention mechanism has also been investigated for sentiment analysis [36]. For instance, Yang et al. [33] propose a hierarchical attentive model which can pick out important features in both word-level and sentence-level. [28] and [5] treat the global user preferences and product characteristics as attention guiders and have brought a nice performance gain. Ma et al. [15] try to model multiple objects discussed in one sentence at one time.

In spite of the success of these sequence-structured RNN methods discussed above, the syntactic hierarchy among the words in a sentence is neglected. [34] and [22] show that linguistic structure has obvious benefits for tasks with highly-formalized outputs such as code generation and semantic parsing. Chen et al. [4] improve neural machine translation using a syntax-aware encoder and decoder, and the improvement is greater for longer sentences. As a result, NLP should re-embrace linguistic structure. It's time to announce that linguistic structure is coming back. Motivated by these recent great researches, we design a hierarchical architecture where in word level each sentence of a review is modeled using an attention-based dependency tree-LSTM, and in sentence level an attention-based long short-term memory network (LSTM) is utilized to generate the review embedding for final sentiment classification.

## 5 Conclusions

With the trend that the demands from people for multimedia analysis is becoming more variant and broad, the Internet-of-Things (IoT) has shown a promising prospect. Motivated by the renaissance of linguistic structure, in this paper, we present a novel hierarchical architecture to deal with review sentiment classification. Taking the dependency parsing tree of sentences, we first encode each sentence of a review using an attention-based dependency tree-LSTM. Then in sentence level, an attention-based LSTM is used to generate the review embedding for final sentiment classification. Finally, we evaluate the architecture on three real-world datasets and verify its superiority over other baselines.

Although tree-structured LSTMs are capable of learning better representations depending on syntactic information than sequential LSTMs, they suffer from the underfitting issue: the same shared compositional function throughout the whole compositional procedure results in the lack of expressive power. In the future, to further improve the sentiment classification performance, we will make our efforts to enhance the expressive power of tree-structured neural networks.

**Acknowledgements.** This work was supported in part by the National Natural Science Foundation of China under grants No. 61821003 and the National Key Research and Development Program of China under grant No. 2016YFB0800402.

## References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014). http://arxiv.org/abs/1409.0473
2. Bhatia, P., Ji, Y., Eisenstein, J.: Better document-level sentiment analysis from RST discourse parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, 17–21 September 2015, Lisbon, Portugal, pp. 2212–2218 (2015)
3. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 740–750 (2014)
4. Chen, H., Huang, S., Chiang, D., Chen, J.: Improved neural machine translation with a syntax-aware encoder and decoder. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, vol. 1, pp. 1936–1945 (2017)
5. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, 1–4 November 2016, Austin, Texas, USA, pp. 1650–1659 (2016)
6. Cheng, K., Li, J., Tang, J., Liu, H.: Unsupervised sentiment analysis with signed social networks. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4–9 February 2017, San Francisco, California, USA, pp. 3429–3435 (2017)
7. Diao, Q., Qiu, M., Wu, C., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, 24–27 August 2014, New York, NY, USA, pp. 193–202 (2014)
8. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, 22–27 June 2014, Baltimore, MD, USA, vol. 2, pp. 49–54 (2014)
9. Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: DRAW: a recurrent neural network for image generation. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 6–11 July 2015, Lille, France, pp. 1462–1471 (2015)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751 (2014)
12. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014, pp. 1188–1196 (2014)
13. Liu, Y., et al.: Deep self-taught hashing for image retrieval. IEEE Trans. Cybern. **49**(6), 2229–2241 (2019)

14. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, 17–21 September 2015, Lisbon, Portugal, pp. 1412–1421 (2015)
15. Ma, X., Zeng, J., Peng, L., Fortino, G., Zhang, Y.: Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis. Future Gen. Comput. Syst. **93**, 304–311 (2018)
16. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, pp. 1045–1048 (2010)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013)
18. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December 2014, Montreal, Quebec, Canada, pp. 2204–2212 (2014)
19. Mullen, T., Collier, N.: Sentiment analysis using support vector machines with diverse information sources. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, Held in Conjunction with ACL 2004, 25–26 July 2004, Barcelona, Spain, pp. 412–418 (2004)
20. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, 6–7 July 2002, Philadelphia, PA, USA, vol. 10, pp. 79–86 (2002)
21. Pérez-Rosas, V., Banea, C., Mihalcea, R.: Learning sentiment lexicons in Spanish. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, pp. 3077–3081 (2012)
22. Rabinovich, M., Stern, M., Klein, D.: Abstract syntax networks for code generation and semantic parsing. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, 30 July–4 August, Vancouver, Canada, vol. 1, pp. 1139–1149 (2017)
23. Rao, D., Ravichandran, D.: Semi-supervised polarity lexicon induction. In: 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, EACL 2009, 30 March–3 April 2009, Athens, Greece, pp. 675–682 (2009)
24. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kociský, T., Blunsom, P.: Reasoning about entailment with neural attention. CoRR abs/1509.06664 (2015), http://arxiv.org/abs/1509.06664
25. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, pp. 1631–1642 (2013)

26. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, 26–31 July 2015, Beijing, China, vol. 1, pp. 1556–1566 (2015)
27. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, 17–21 September 2015, Lisbon, Portugal, pp. 1422–1432 (2015)
28. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, 26–31 July 2015, Beijing, China, vol. 1, pp. 1014–1023 (2015)
29. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, 22–27 June 2014, Baltimore, MD, USA, vol. 1, pp. 1555–1565 (2014)
30. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, 1–4 November 2016, Austin, Texas, USA, pp. 606–615 (2016)
31. Xu, J., Chen, D., Qiu, X., Huang, X.: Cached long short-term memory neural networks for document-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, 1–4 November 2016, Austin, Texas, USA, pp. 1660–1669 (2016)
32. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 6–11 July 2015, Lille, France, pp. 2048–2057 (2015)
33. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 12–17 June 2016, San Diego California, USA, pp. 1480–1489 (2016)
34. Yin, P., Neubig, G.: A syntactic neural model for general-purpose code generation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, 30 July–4 August, Vancouver, Canada, vol. 1, pp. 440–450 (2017)
35. Zeng, J., Ma, X., Zhou, K.: CAAE++: improved CAAE for age progression/regression. IEEE Access **6**, 66715–66722 (2018)
36. Zeng, J., Ma, X., Zhou, K.: Enhancing attention-based LSTM with position context for aspect-level sentiment classification. IEEE Access **7**, 20462–20471 (2019)
37. Zhao, J., Gui, X.: Deep convolution neural networks for Twitter sentiment analysis. IEEE Access (2017)
38. Zhou, K., Zeng, J., Liu, Y., Zou, F.: Deep sentiment hashing for text retrieval in social CIoT. Future Gen. Comput. Syst. **86**, 362–371 (2018)